Thiago Akinori Ykeda Sato

Seleção de elementos rotulados para o aprendizado semissupervisionado baseado em grafos

São José dos Campos

02 de Outubro de 2020

Thiago Akinori Ykeda Sato

Seleção de elementos rotulados para o aprendizado semissupervisionado baseado em grafos

Projeto apresentado ao curso de Engenharia de Computação da Universidade Federal de São Paulo como parte do requisito para a aprovação na disciplina Elaboração de Trabalhos Científicos e Tecnológicos em Computação.

UNIVERSIDADE FEDERAL DE SÃO PAULO

Departamento de Ciência e Tecnologia

Engenharia de Computação

Orientador: Prof[a]. Dr[a]. Lilian Berton

São José dos Campos

02 de Outubro de 2020

Elaborado por sistema de geração automática com os dados fornecidos pelo(a) autor(a).

Ykeda Sato, Thiago Akinori
Seleção de elementos rotulados para o aprendizado semissupervisionado baseado em grafos/ Thiago Akinori Ykeda Sato
Orientador(a) Lilian Berton-São José dos Campos, 2020.
68 p.

Trabalho de Conclusão de Curso-Engenharia da Computação-Universidade Federal de São Paulo-Instituto de Ciência e Tecnologia, 2020.

1. Aprendizado Semissupervisionado. 2. Grafo. 3. Medidas de Centralidade. I. Berton, Lilian , orientador(a). II. Título.

Thiago Akinori Ykeda Sato

Seleção de elementos rotulados para o aprendizado semissupervisionado baseado em grafos

Projeto apresentado ao curso de Engenharia de Computação da Universidade Federal de São Paulo como parte do requisito para a aprovação na disciplina Elaboração de Trabalhos Científicos e Tecnológicos em Computação.

Profª. Drª. Lilian Berton
Orientador

Dr. Roberto Gueleri
Convidado 1

Dr. Didier Vega Oliveros
Convidado 2

São José dos Campos
02 de Outubro de 2020

Este trabalho é dedicado a todos que me ajudaram na minha trajetória.

Agradecimentos

Agradeço a todos que me apoiaram à completar essa fase da minha vida. Em especial, minha família que com muito empenho me deu suporte financeiro durante parte dessa jornada, além de suporte emocional para aguentar todos esses anos de esforço. Finalmente, agradeço a todos os professores que, com tanto empenho, me deram o conhecimento necessário para essa vitória.

"Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)

Resumo

A escassez de dados rotulados tem aumentado o interesse no aprendizado semissupervisionado, o qual emprega uma proporção pequena de dados rotulados e uma proporção grande de dados não-rotulados para realizar classificação em grandes bases de dados. Este trabalho tem como objetivo analisar a influência de elementos rotulados no aprendizado semissupervisionado. Buscamos melhorar a performance de modelos semissupervisionados baseados em grafos a partir da seleção de elementos rotulados. As seleções foram baseadas na importância do nó dentro do grafo, utilizando-se de medidas de centralidade, dentre essas medidas, a métrica *betweeness* teve destaque. Também analisamos a distribuição de elementos rotulados por comunidades e notamos que quando essa distribuição é balanceada há um aumento na acurácia.

Palavras-chave: Classificação de Dados. Aprendizado de Máquina. Aprendizado Semissupervisionado. Métodos baseados em Grafos. Medidas de Centralidade. Comunidades.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Atualmente, a tecnologia evoluiu de forma que a quantidade de dados digitais coletados começou a crescer rapidamente (CHEN et al., 2013). De acordo com previsões de 2010, vamos alcançar 149 zettabytes de dados gerados, copiados e consumidos no mundo em 2024 (HOLST, 2020). Um dos motivos pelo qual esse volume cresce tanto é a facilidade de armazenamento e o interesse em utilizar esse dados para melhorar o processo de tomada de decisões (L'HEUREUX et al., 2017).

Algoritmos de aprendizado de máquina estão entre os métodos mais utilizados para essa tomada de decisões, porém, algoritmos tradicionais foram desenvolvidos antes de termos um volume de dados tão grande (L'HEUREUX et al., 2017), para isso, novos paradigmas de processamento de dados e novos algoritmos estão sendo desenvolvidos. Geralmente, a literatura divide os algoritmos em: supervisionados e não supervisionados. Enquanto que os não supervisionados visam encontrar uma descrição a partir de um conjunto de dados, os algoritmos supervisionados induzem hipóteses a partir de um conjunto de exemplos para prever novos exemplos (FACELI et al., 2011).

Recentemente, um novo paradigma foi desenvolvido: o aprendizado semissupervisionado, híbrido entre os tipos anteriores, no qual, além dos dados não rotulados, o algoritmo é provido de alguns dados rotulados (CHAPELLE; SCHOLKOPF; ZIEN, 2009). A sua maior vantagem é utilizar tanto dados rotulados quanto dados não rotulados para atingir uma melhor performance que algoritmos supervisionados (ZHU; GOLDBERG, 2009). Considerando o grande volume de dados disponíveis, pode ser custoso e trabalhoso, rotular grandes quantidades de dados. Além disso, existem casos, como prever doenças em pacientes, que demandam especialistas no assunto o que pode dificultar o processo ainda mais.

Alguns paradigmas tentam selecionar os elementos mais representativos para ser rotulados como *Active Learning*, no qual, um oráculo (geralmente um usuário) é questionado sobre o rótulo de um exemplo não rotulado para melhorar a performance de um algoritmo de aprendizado diminuindo o volume do conjunto de dados utilizado como entrada (SETTLES, 2009). Por isso, é interessante melhorar a qualidade de dados utilizados como entrada em um algoritmo semissupervisionado.

Nesse trabalho, será utilizado uma representação dos dados em forma de grafos e serão empregadas medidas de centralidade para selecionar os pontos (ou exemplos) mais centrais para serem rotulados. Assim, espera-se uma melhor performance de algoritmos semissupervisionados baseados em grafos do que uma seleção aleatória de rótulos.

2 Objetivos

2.1 Objetivo Geral

O objetivo desse trabalho é melhorar a classificação de modelos semissupervisionados a partir de uma seleção mais eficaz de elementos rotulados.

Como o aprendizado semissupervisionado faz uso de poucos dados rotulados na classificação, a hipótese do trabalho é de que a seleção de elementos mais representativos para rotulagem melhore a acurácia dos algoritmos.

2.2 Objetivos Específicos

- Identificar elementos representativos no grafo para rotulagem a priori, por meio de medidas de centralidade.
- Realizar um comparativo entre técnicas de rotulagem a fim de identificar quais delas fornecem um ganho de acurácia na classificação semissupervisionada.
- Analisar a distribuição de rótulos por comunidades.

2.3 Organização do Documento

Esse documento está organizado do seguinte modo: no Capítulo 3 é apresentada a fundamentação teórica, com os principais conceitos usados no trabalho, como aprendizado de máquina, métricas de avaliação, aprendizado semissupervisionado, construção de grafos e medidas de centralidade. No Capítulo 4 é apresentado alguns trabalhos relacionados. No Capítulo 5 é apresentada a metodologia para realização do trabalho, além de *datasets* e bibliotecas. No Capítulo 6 os resultados encontrados. Por fim, no Capítulo 7 as considerações finais.

3 Fundamentação Teórica

3.1 Aprendizado de Máquina

Aprendizado de Máquina (*Machine Learning* em inglês) é o campo de Inteligência Artificial que "visa construir programas de computadores capazes de melhorar com a experiência automaticamente" (MITCHELL, 1997). Porém, antes de detalhar os conceitos principais dessa área, algumas definições são dadas a seguir de acordo com (MONARD; BARANAUSKAS, 2003):

- Indutor: gera uma hipótese a partir de um conjunto de dados;
- Exemplo: também conhecido como dado ou caso, é um conjunto de valores que descreve um objeto de interesse;
- Atributo: descreve alguma característica ou informação de um exemplo;
- Classe: ou rótulo, é o atributo especial que descreve o evento que será predito;
- Conjunto de exemplos: ou conjunto de dados, contém os exemplos com atributos e rótulos associados.

Na Tabela 1, pode-se observar um exemplo de conjunto de dados, na qual, cada linha e_j representa um dado, ou exemplo, e cada coluna x_i representa um atributo. O atributo que é o interesse em ser classificado é a classe, por exemplo, pode ser escolhida a coluna x_n como rótulo.

Tabela 1 – Exemplo de conjunto de dados

	x_1	...	x_i	...	x_n
e_1	a_{11}	...	a_{i1}	...	a_{n1}
...	...		...		...
e_j	a_{1j}	...	a_{ij}	...	a_{nj}
...	...		...		...
e_m	a_{1m}	...	a_{im}	...	a_{nm}

De acordo com FACELI et al. (2011), os algoritmos de Aprendizado de Máquina induzem hipóteses, ou aproximações de funções, a partir de um conjunto de dados. A área pode ser divida em dois paradigmas principais:

- Predição, ou supervisionados;

- Descrição, ou não supervisionados.

Algoritmos supervisionados recebem um conjunto de exemplos com valores, ou atributos e os rótulos de cada classe, sendo esses discretos ou contínuos, e criam hipóteses que predizem exemplos não rotulados. Enquanto que algoritmos não supervisionados, a hipótese separa esses valores em grupos com características comuns (MONARD; BARANAUSKAS, 2003). Outra abordagem mais recente são os algoritmos semissupervisionados, que fazem uso de muitos elementos não rotulados e poucos elementos rotulados (CHAPELLE; SCHOLKOPF; ZIEN, 2009).

3.2 Métricas de Avaliação de Modelos Preditivos

3.2.1 Importância e Dificuldades na Avaliação

A precisão da avaliação de acurácia em modelos de aprendizado de máquina é importante para, por exemplo, escolher a hipótese a ser utilizada, ou, ainda muitos métodos usam da avaliação como parte integral do modelo. Estimar a acurácia de um modelo é relativamente simples quando há um conjunto grande de dados, porém, quando o conjunto é limitado aparece duas dificuldades (MITCHELL, 1997):

- Viés na estimativa: A acurácia observada em um modelo de aprendizado sobre os exemplos de treino é, frequentemente, uma estimativa fraca, já que, a hipótese foi derivada desse conjunto.
- Variação na estimativa: A acurácia de teste pode variar mesmo em um conjunto sem viés, dependendo da formação de exemplos do conjunto de teste.

3.2.2 Métrica de Erro

Usualmente, é empregado uma taxa de erro, ou classificações incorretas, na qual, considerando um caso binário, $I(a) = 1$, se a é verdade e $I(a) = 0$, caso contrário. Assim, se considerarmos uma classe predita, $\hat{f}(x_i)$, de um exemplo x_i e uma classe conhecida de x_i, y_i, para calcular o erro temos (FACELI et al., 2011):

$$err(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{f}(x_i)) \tag{3.1}$$

Com n, sendo o tamanho do conjunto e $0 \leq err(\hat{f}) \leq 1$. Quanto mais próximo de 0, melhor a taxa de erro do modelo, pois maximiza a taxa de acerto ou acurácia, dada por FACELI et al. (2011):

$$ac(\hat{f}) = 1 - err(\hat{f}) \tag{3.2}$$

3.2.3 Matriz de Confusão

A matriz de confusão tem como objetivo ilustrar o desempenho do classificador ao demonstrar medidas quantitativas de quais classes o algoritmo acerta ou erra mais. As linhas da matriz representam as classes reais do conjunto e as colunas representam as classes preditas pelo modelo sendo avaliado. Dada uma matriz de confusão M_c, onde cada elemento linha i e cada coluna j representa o número de exemplos da classe i preditos como classe j.

Logo, para cada classe k, a matriz possui uma dimensão de $k \times k$. Assim cada elemento onde $i = j$ mostra o volume de acerto do classificador para a classe i, e cada elemento onde $i \neq j$ mostra o volume de erros no qual a classificação de j deveria ser i (FACELI et al., 2011). Pode ser observado um exemplo na Tabela 3:

Tabela 2 – Exemplo de uma matriz de confusão

	classe 1	classe 2	classe 3
classe 1	11	1	3
classe 2	1	4	0
classe 3	2	1	6

Fonte: (FACELI et al., 2011)

3.2.4 Medidas de Desempenho

Simplificando o problema para duas classes onde uma classe é chamada de positiva (+) e outra, de negativa (-), então considerando uma matriz de confusão para o resultado do modelo na qual:

- VP (Verdadeiro Positivo): quantidade de exemplos classificados corretamente como positivos;
- VN (Verdadeiro Negativo): quantidade de exemplos classificados corretamente como negativo;
- FP (Falso Positivo): quantidade de exemplos classificados falsamente como positivos;
- FN (Falso Negativo): quantidade de exemplos classificados falsamento como negativos;

Onde a somatória de todos os elementos da matriz, representada abaixo, resulta no total de elementos do conjunto de dados.

Considerando os conceitos acimas e o problema de duas classes temos as seguintes métricas:

Tabela 3 – Matriz de confusão para duas classes

	+	-
+	VP	FN
-	FP	VN

Fonte: (FACELI et al., 2011)

- *Precision* (ou Precisão, em português): a métrica de *Precision* é a proporção de positivos que foram classificados corretamente pelo modelo $\hat{f}$ sobre todos os classificados como positivos (FACELI et al., 2011), ou seja:

$$prec(\hat{f}) = \frac{VP}{VP + FP} \tag{3.3}$$

  O *precision* pode ser entendido como a métrica de exatidão do modelo, ou o quanto exemplos da classe positiva realmente pertencem a classe positiva (FACELI et al., 2011).

- *Recall* (ou Revocação, em português): a métrica *Recall* representa a taxa de acerto da classe positiva (FACELI et al., 2011), na qual:

$$rec(\hat{f}) = \frac{VP}{VP + FN} \tag{3.4}$$

  O *recall* pode ser entendido como a métrica de completude do modelo ou o quanto exemplos da classe positiva foram classificados como sendo positivos (FACELI et al., 2011).

Todas as medidas citadas acima podem ser generalizadas para problemas multi-classes, que possuem mais de uma classe, assumindo uma estratégia um contra todos onde cada classe pode ser considerada como a classe positiva e as outras em conjunto como a classe negativa (FACELI et al., 2011).

Em geral, as métricas de *precision* ou *recall* não são utilizadas separadas, e sim, combinadas em uma média harmônica ponderada por um peso w, onde:

$$F_m(\hat{f}) = \frac{(w+1) \times rec(\hat{f}) \times prec(\hat{f})}{rec(\hat{f}) + w \times prec(\hat{f})} \tag{3.5}$$

Utilizando a métrica F_m como $w = 1$ temos:

$$F_1(\hat{f}) = \frac{2 \times rec(\hat{f}) \times prec(\hat{f})}{rec(\hat{f}) + prec(\hat{f})} \tag{3.6}$$

Assim, a métrica F_1 pode que ser entendida como uma média entre as métricas de *precision* ou *recall*.

Em adição às medidas descritas acimas, temos outras que dizem respeito aos seguintes aspectos adicionais (HAN; PEI; KAMBER, 2011):

- Velocidade: custo computacional envolvido no cálculo;
- Robustez: a acurácia em casos de classificações com dados faltando ou com ruído;
- Escalabilidade: construção de modelos eficientes dado um conjunto grande de dados;
- Interpretabilidade: medida subjetiva que diz respeito ao nível de entendimento que é dado pelo preditor;

3.2.5 Amostragem

Geralmente, o conjunto de dados deve ser utilizado tanto para a criação do indutor quanto para a avaliação. Utilizar-se dos mesmos dados para treinar e avaliar o modelo, pode levar à uma estimativa otimista, já que algoritmos de aprendizado utilizam dos exemplos para melhorar o desempenho. O uso do mesmo conjunto de exemplos no treinamento e na avaliação é conhecido como resubstituição (FACELI et al., 2011).

Deve-se então utilizar de métodos de amostragem para obter estimativas de desempenho mais confiáveis, os principais tipos de métodos são:

- Método *Holdout* e Amostragem aleatória: O método divide o conjunto aleatoriamente em dois, um subconjunto de treinamento e outro de avaliação, enquanto que a amostragem aleatória realiza o método *holdout* k vezes e o valor final é a média de cada iteração (HAN; PEI; KAMBER, 2011);
- *Cross Validation*, ou Validação Cruzada: Conhecido também como *k-fold cross-validation*, o conjunto inicial é dividido em k subconjuntos mutuamente exclusivos $(D_1, D_2, .., D_k)$ aproximadamente do mesmo tamanho. Em cada iteração i, o subconjunto D_i é utilizado como conjunto de avaliação do modelo de aprendizado e os outros como conjunto de treino;

3.3 Aprendizado Semissupervisionado

O paradigma de aprendizado semissupervisionado é um híbrido entre o aprendizado supervisionado e o não supervisionado, tanto que a maioria das estratégias de aprendizado semissupervisionado se apoia em algum dos dois paradigmas (ZHU; GOLDBERG, 2009). Além de dados sem rótulos (característico do aprendizado não supervisionado), o algoritmo é provisionado de alguns dados com rótulos. O conjunto de dados X, onde $X := (x_1, x_2, ..., x_n)$, pode ser divido em dois conjunto: o primeiro $X_l := (x_1, x_2, ..., x_l)$, no qual os rótulos são $Y_l := (y_1, y_2, ..., y_n)$, e o segundo $X_u := (x_{l+1}, x_{l+2}, ..., x_n)$ sem rótulos (CHAPELLE; SCHOLKOPF; ZIEN, 2009).

Para que o algoritmo obtenha um resultado satisfatório, deve ser considerado a distribuição dos exemplos no conjunto de dados, nos quais os dados não rotulados vão ajudar a encontrar uma solução relevante para o problema de classificação. Ou seja, para que o conhecimento em $p(x)$ possa ser inferido em $p(y|x)$ existem três suposições (CHAPELLE; SCHOLKOPF; ZIEN, 2009):

- Suposição da suavidade: se dois pontos x_1 e x_2 estão próximos em uma região de alta densidade, então os rótulos y_1 e y_2 também devem ser próximos;
- Suposição de agrupamento: se dois pontos estão no mesmo agrupamento, provavelmente são da mesma classe.
- Suposição de *manifold*: um dado de alta dimensionalidade está aproximadamente em uma baixa dimensão de *manifold*.

3.4 Construção de grafos

Muitos algoritmos semissupervisionados se baseiam na geometria dos dados induzidas de exemplos rotulados e não rotulados. Uma das possibilidades é representar a geometria a partir de um grafo empírico $G = (V, E)$ onde os nós $V = \{x_1, \ldots, x_n\}$ representam o conjunto de treinamento e as arestas E representam as similaridades entre cada vértice.

Essas similaridades são dadas por uma matriz de pesos $W : W_{ij}$, onde $W_{ij} \neq 0$ se x_i e x_j são vizinhos. Um exemplo de matriz de pesos é a *k-nearest neighbor* matrix (ou matriz de k vizinhos mais próximos): $W_{ij} = 1$ se x_i está entre os k vizinhos mais próximos de x_j ou vice-versa, entretanto, se não estiver será zero. Outra matriz de pesos comum é dada pelo kernel Gaussiano de largura σ (CHAPELLE; SCHOLKOPF; ZIEN, 2009):

$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \tag{3.7}$$

Outras medidas de similaridade que também podem ser empregadas são a distância Minkowski, Euclidiana etc.

3.4.1 Distância de Minkowski

A distância de Minkowski é a base para outra medidas como a Euclidiana. A medida de ordem p, onde p é um inteiro e define as variações dessa métrica, entre dois pontos $X_i = (x_{i1}, x_{i2}, ..., x_{in})$ e $X_j = (x_{j1}, x_{j2}, ..., x_{jn})$ é definida por FACELI et al. (2011):

$$D(X_i, X_j) = (\sum_{k=1}^{n} |x_i^k - x_j^k|^p)^{\frac{1}{p}} \tag{3.8}$$

3.4.2 Distância Euclidiana

Definida pela Distância de Minkowski na qual $p = 2$, assim é possível obter (FACELI et al., 2011):

$$D(X_i, X_j) = \sqrt{\sum_{k=1}^{n} (x_i^k - x_j^k)^2} \tag{3.9}$$

3.5 Métodos de Classificação baseados em Grafos

O mais comum nesses métodos é que os exemplos do conjunto de dados sejam representados como nós de um grafo e as arestas representam a distância entre um par de nós. Se a distância de dois pontos é calculada minimizando a distância do caminho agregando todos os caminhos que conectam os dois pontos, isso pode ser visto como uma aproximação da distância geodésica dos dois pontos em relação à variedade dos pontos de dados (CHAPELLE; SCHOLKOPF; ZIEN, 2009).

Considerando $G = (V, E)$, um grafo com arestas de pesos reais dado por $\omega : E \to \Re$. Aqui o peso $\omega(e)$ de uma aresta e indica a semelhança dos nós incidentes (e uma aresta ausente na matriz de pesos corresponde a zero semelhança). A matriz de adjacência do grafo G pode ser dada por CHAPELLE; SCHOLKOPF; ZIEN (2009):

$$W_{i,j} := \begin{cases} w(e), e = (i,j) \in E, \\ 0, e = (i,j) \notin E \end{cases} \tag{3.10}$$

Normalmente, a predição dessa estratégia consiste em rótulos para dados não rotulados, ou seja, essa estratégia é intrinsecamente transdutiva, já que retorna o valor da função de decisão e não a própria função (CHAPELLE; SCHOLKOPF; ZIEN, 2009).

3.5.1 *Label Propagation* (LP)

Dado que $\{(x_1, y_1), \ldots, (x_l, y_l)\}$ seja um conjunto de dados com rótulos, onde $Y_l = \{y_1, \ldots, y_l\}$ são rótulos das os elementos do conjunto e assumindo que o número de classes C é conhecido e que todas as classes estejam presentes no conjunto de dados rotulados. Assim, dado que $\{(x_{l+1}, y_{l+1}), \ldots, (x_u, y_u)\}$ seja um conjunto de dados sem rótulos, onde $Y_l = \{y_l, \ldots, y_n\}$ não foi observado. Sabendo que $X = (x_1, \ldots, x_{l+u}) \in R^D$ (ZHU; GHAHRAMANI, 2002).

O objetivo é que pontos próximos possuam rótulos similares, para isso, é criado um grafo totalmente conexo onde todos os nós são pontos do conjunto de dados rotulados ou não. As arestas entre quaisquer nós i, j possuem pesos tal que quanto mais próximos

os nós estão utilizando Distância Euclidiana, maior peso $\omega_{i,j}$, os pesos são controlados por um parâmetro σ (ZHU; GHAHRAMANI, 2002):

$$\omega_{i,j} = exp(-\frac{d_{i,j}^2}{\sigma^2}) \tag{3.11}$$

É possível utilizar outras métricas de distância para que o algoritmo seja mais apropriado para o dado x. Todos os nós possuem rótulos leves que podem ser interpretados como distribuições sobre os rótulos. Os rótulos são propagados para todos os nós a partir das arestas. É definido uma matriz $(l+u) \times (l+u)$ de transição probabilística T (ZHU; GHAHRAMANI, 2002):

$$T_{i,j} = P(j \rightarrow i) = \frac{\omega_{i,j}}{\sum_{k=1}^{l+u} \omega_{kj}} \tag{3.12}$$

Onde $T_{i,j}$ é a probabilidade do salto do nó j para o i. Também é definida uma matriz Y de rótulos $(l+u) \times C$ onde a i-ésima linha representa a distribuição de probabilidade do rótulo de um nó x_i (ZHU; GHAHRAMANI, 2002). O algoritmo pode ser resumido como:

1. Propagar $Y \leftarrow TY$.
2. Normalizar linha Y.
3. Fixar o dados rotulados e repetir o passo 1 até Y convergir.

No passo 1, todos os nós propagam os seus rótulos. O passo 2, serve para que Y mantenha a interpretação de probabilidade dos rótulos. E no passo 3, se mantém a persistência dos dados rotulados (ZHU; GHAHRAMANI, 2002).

3.5.2 *Local and Global Consistency* (LGC)

Considerando o mesmo problema do algoritmo anterior, o objetivo do LGC é também predizer rótulos de exemplos não rotulados. Dado um conjunto de pontos $X = \{x_1, \ldots, x_l, x_{l+1}, \ldots, x_n\} \subset \Re^m$ e um conjunto de rótulos $L = \{1, \ldots, c\}$, os primeiros l pontos x_i $(i \leq l)$ são rotulados como $y_i \in L$ e os pontos restantes x_u $(l+1 \leq u \leq n)$ são não rotulados e serão rotulados pelo algoritmo (ZHOU et al., 2004).

Com φ denotando um conjunto de $n \times c$ matrizes com entradas não negativas, uma matriz $F = [F_1^T, ..., F_n^T]^T \in \varphi$ corresponde a uma classificação do conjunto de dados X rotulando cada ponto x_i como um rótulo $y = argmax_{j \leq c} F_{ij}$. F pode ser interpretado como uma função vetorial $F : X \rightarrow \Re^c$ na qual atribui um vetor F_i para cada ponto x_i. A matriz $n \times c$ chamada Y, onde $Y \in \varphi$ com $Y_{ij} = 1$ se x_{ij} é rotulado como $y_i = j$ e $Y_{ij} = 0$, caso contrário (ZHOU et al., 2004). O algoritmo é definido por:

1. Formar a matriz de afinidade W definida por $W_{ij} = exp(-\|x_i - x_j\|^2/2\sigma^2)$ se $i \neq j$ e $W_{ii} = 0$.

2. Construir a matriz $S = D^{-1/2}WD^{-1/2}$, onde D é a matriz diagonal com o (i,i)-ésimo elemento iguais a soma da i-ésima linha de W.

3. Itere $F(t+1) = \alpha SF(t) + (1-\alpha)Y$ até a convergência, onde α é um parâmetro entre (0,1).

4. Com F^* denotando o limite da sequência $F(t)$, rotular cada ponto x_i como rótulo $y_i = argmax_{j \leq c} F^*_{ij}$.

Primeiro, W foi definido como uma relação entre pares no conjunto de dados X com a diagonal sendo zero. O grafo $G = (V, E)$ pode ser definido em X, onde os nós em V são definidos por X e as arestas E têm o peso definido por W. No segundo passo a matriz de pesos W é normalizada simetricamente, e durante a iteração do terceiro passo cada ponto recebe a informação dos seus vizinhos e mantém sua própria informação inicial. Por fim, o rótulo de cada ponto não rotulado é definido para ser a classe na qual recebeu mais informação durante o processo (ZHOU et al., 2004).

3.5.3 *Harmonic Functions* (HMN)

Supondo que há l elementos rotulados, $\{(x_1, y_1), ..., (x_l, y_l)\}$, e u elementos não rotulados, $(x_{u+1}, y_{u+1}), ..., (x_{l+u}, y_{l+u})$, onde, geralmente, $l \ll u$, e n o número de elementos é dado por: $n = l + u$. Considerando os rótulos em $y \in \{0, 1\}$, temos um grafo conexo $G(V, E)$, onde V corresponde aos n elementos que podem ser divididos nos subconjuntos de nós $L = \{1, ..., l\}$ e $U = \{l + 1, ..., u\}$ nos quais L corresponde aos elementos rotulados $y_1, ..., y_l$ e U aos elementos não rotulados $y_{l+1}, ..., y_{l+u}$, o objetivo é dar rótulos para os elementos em U (ZHU; GHAHRAMANI; LAFFERTY, 2003).

Definindo uma matriz $n \times n$ de afinidade W, quando $x \in \Re^m$, W é dado por:

$$w_{ij} = exp(-\sum_{d=1}^{m} \frac{(x_{id} - x_{jd})^2}{\delta_d^2}) \tag{3.13}$$

A rotulação do elementos será dada pela função f que é computada por $f : V \to \Re$ em G com certas propriedades. Inicialmente, f deve tomar os valores $f(i) = f_l(i) \equiv y_i$, onde i são os nós rotulados em L. Para que os nós não rotulados próximos no grafo possuam o mesmo rótulo, a função de energia quadrática foi escolhida:

$$E(f) = \frac{1}{2} \sum w_{ij}(f(i) - f(j))^2 \tag{3.14}$$

Campos Gaussianos, dados por $p_\beta(f) = \frac{e^{-\beta E(f)}}{Z_\beta}$, são formados para atribuir distribuições de probabilidade nas funções f, onde β é um parâmetro de temperatura inversa e

Z_β é uma função de partição $Z_\beta = \int_{f|L=1} exp(-\beta E(f))df$ que normaliza todas as funções para f_l nos dados rotulados (ZHU; GHAHRAMANI; LAFFERTY, 2003).

Como a função $f = argmin_{f|L=f_l} E(f)$ é harmônica (ZHU; GHAHRAMANI; LAFFERTY, 2003), o valor de f em cada elemento j não rotulado é a média de f nos nós vizinhos, ou seja:

$$f(j) = \frac{1}{d_j} \sum_{i\ j} w_{ij} f(i), \text{ para } j = l+1, ..., l+u \tag{3.15}$$

A função acima pode ser escrita em termos matriciais dividindo a matriz de afinidade W em 4 blocos após a l-ésima linha e coluna, e considerando a diagonal D e $P = D^{-1}W$ similarmente separados, temos:

$$W = \begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix} \tag{3.16}$$

sendo a função $f = \begin{bmatrix} f_l \\ fu \end{bmatrix}$, na qual f_u representa os elementos não rotulados, e sabendo pela propriedade harmônica que $\Delta f = 0$, temos:

$$f_u = (D_{uu} - W_{uu})^{-1} W_{ul} f_l = (I - P_{uu})^{-1} P_{ul} f_l \tag{3.17}$$

onde, f_u é a base para esse método de aprendizado semissupervisionado (ZHU; GHAHRAMANI; LAFFERTY, 2003).

3.6 Medidas de Centralidade

Um grafo G é formado por um conjunto de vértices V e arestas E, pode ser representado por $G(V, E)$. Com relação a representação computacional, pode ser representado tanto como uma lista de adjacência quanto por uma matriz de adjacência (CORMEN et al., 2009).

Dado um grafo é possível traçar um caminho, o qual é definido como qualquer sequência de vértices de modo que cada par consecutivo da sequência é conectada por arestas (NEWMAN, 2010). O tamanho de um caminho é o número de arestas contidas no caminho, e, por meio de algoritmos como Busca em Largura (CORMEN et al., 2009), é possível encontrar o menor caminho / caminho mínimo entre dois vértices, o qual possui o menor tamanho entre todos os possíveis caminhos (CORMEN et al., 2009).

Dentro de um grafo, é possível calcular uma variedade de medidas para quantificar a topologia da rede. Um dos conceitos passíveis para este fim é a centralidade, a qual visa encontrar os nós mais importantes ou centrais (NEWMAN, 2010).

3.6.1 Centralidade de Grau

Possivelmente, a centralidade de *Grau* é a métrica mais simples de centralidade, na qual, é descrita como o número de arestas conectadas em um vértice. Para um vértice i denotamos o grau dele como k_i, considerando um grafo não direcionado de n vértices representado por uma matriz de adjacência A (NEWMAN, 2010), temos:

$$k_i = \sum_{j=1}^{n} A_{ij} \tag{3.18}$$

E a complexidade computacional de calcular a centralidade de grau para todos os vértices do grafo, considerando cada aresta em ambos os vértices conectados, é de $O(V)$ com V sendo o número de nós no grafo (GRANDO; NOBLE; LAMB, 2016).

3.6.2 Centralidade *Closeness*

A centralidade *Closeness* é a média da menor distância entre um vértice i e os outros vértices da rede. Supondo que d_{ij} é a menor distância do caminho entre i e j, temos (NEWMAN, 2010):

$$l_i = \frac{1}{n} \sum_{j} d(i,j) \tag{3.19}$$

A complexidade computacional para calcular o *closeness* para todos os nós do grafo é de $O(|V||E|)$, com V sendo o número de nós no grafo e E sendo o número de arestas (GRANDO; NOBLE; LAMB, 2016).

3.6.3 Centralidade *Betweenness*

A centralidade *Betweenness* mede o quanto um vértice está posicionado no caminho de outros vértices, assim um nó com alta centralidade *Betweenness* possui uma considerável influência dentro de um grafo já que muitos caminhos passam por tal vértice. Assim, considerando $\eta_{st}^i = 1$ se o vértice está no caminho mínimo entre os nós s e t, e $\eta_{st}^i = 0$ caso contrário, então *Betweenness* b_i de um vértice i é dado por NEWMAN (2010):

$$b_i = \sum_{st} \eta_{st}^i \tag{3.20}$$

Para essa centralidade, a complexidade computacional para todos os nós do grafo é de $O(|V||E|)$, com V sendo o número de nós no grafo e E sendo o número de arestas. Porém, quando arestas com pesos são consideradas, a complexidade aumenta para $O(|V||E| + V^2 \log V)$ (GRANDO; NOBLE; LAMB, 2016)

3.6.4 Coeficiente de Agrupamento

O coeficiente de Agrupamento quantifica o quanto a vizinhança de um nó é conectado, e é definido tanto localmente quanto globalmente (KEMPER, 2009):

- Local: o coeficiente de agrupamento local C_{li}, é a densidade de conexão em um dado nó i, é dado por (KEMPER, 2009):

$$C_{li} = \frac{k}{q} \tag{3.21}$$

  onde k é a quantidade de triângulos conectados no vértice i e q é a quantidade de triplas centrado no nó (KEMPER, 2009). A complexidade computacional depende do grau dos nós do grafo, no pior caso, o custo para calcular o coeficiente para todos os nós do grafos é de $O(V^3)$, com V sendo o número de nós do grafo, e no melhor caso, a complexidade é de $O(V)$ (FRANCESCHET, 2020).

- Global: o coeficiente de agrupamento global C_{gi} é a média do coeficiente local de todos os vértices do grafo, assim (KEMPER, 2009):

$$C_{gi} = \frac{1}{n}\sum_{i} C_i \tag{3.22}$$

3.7 Detecção de Comunidade em Grafos

Em representações de sistemas reais via grafos, uma das análises mais importante é a estrutura de comunidade, ou seja, a organização de vértices em agrupamentos, o qual consiste de muitos vértices conectando-se entre si com vértices do mesmo grupo, e poucos vértices possuindo arestas com vértices de outros grupos (FORTUNATO, 2010). Dentre os vários algoritmos utilizados para detectar comunidade em (FORTUNATO, 2010), o Algoritmo de Louvain (BLONDEL et al., 2008) possui boa performance e complexidade linear.

3.7.1 Algoritmo de Louvain

O algoritmo de Louvain extrai estruturas de comunidades em grandes redes complexas, baseando-se em métodos heurísticos de otimização de modularidade. Assumindo uma rede complexa com N nós, inicialmente todos os nós possuem comunidades diferentes, para cada nó i é calculado o ganho de modularidade ao removê-lo de sua comunidade e inserir na comunidade dos nós vizinhos j, então o nó i é inserido na comunidade que traz maior ganho positivo, se o ganho máximo foi negativo o nó i permanece na própria comunidade. O processo é repetido sequencialmente para todos os nós até que não exista mais ganho.

O ganho de modularidade ΔQ para um nó i de uma comunidade C é calculada por:

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \tag{3.23}$$

onde $\sum_{in}$ é a soma dos pesos das conexões dentro de C, $\sum_{tot}$ é a soma dos pesos das conexões incidentes dos nós em C, k_i é a soma das ligações incidentes ao nó i na comunidade C e m é a soma dos pesos de todas as ligações na rede (BLONDEL et al., 2008).

Em seguida, uma nova rede é criada, na qual os nós são as comunidades e os pesos são calculados a partir da soma dos pesos das conexões entre nós das duas comunidades. Os mesmos processos de cálculo de ganho de modularidade e inserção dos nós são repetidos para esta rede, até que não seja possível juntar nenhuma comunidade à outra (BLONDEL et al., 2008).

4 Revisão da Literatura

O trabalho proposto em PARSAZAD; SABOORI; ALLAHYAR (2012) adapta o algoritmo "aiNet" em YOUNSI; WANG (2004) para invés de agrupar os elementos dos conjuntos de dados, descrevê-los e recomendar os melhores elementos para serem rotulados. Para avaliação dos subconjuntos selecionados, foram usados dois algoritmos semissupervisionados: o *semi-supervised K-Means* e o *semi-supervised Support Vector Machine*. Além de 5 *datasets*: *Iris*, *Soybean*, *Wine*, *Digits-389*, e *Letter-IJL*. Conjuntos aleatórios de rótulos de tamanhos 10, 20, 30 e 40 foram gerados para treinar os algoritmos semissupervisionados com o objetivo de comparar os conjuntos aleatórios com os rótulos selecionados pelo algoritmo proposto. Os resultados demonstraram que o algoritmo de seleção pode melhorar a acurácia dos modelos semissupervionados testados, mas não foi um comportamento que persistiu em todos os *datasets*.

Em PROTOPAPADAKIS; VOULODIMOS; DOULAMIS (2018), algoritmos de seleção de elementos propostos pelos autores a partir de combinações de métodos de amostragem têm os impactos analisados para problemas de reconhecimento de padrões visuais complexos com aprendizado semissupervisionado. Os métodos de amostragem propostos são descritos a seguir:

- *OPTICS (Ordering Points to Identify the Clustering Structure) extrema*: o método aplica o algoritmo *OPTICS* (DASZYKOWSKI; WALCZAK; MASSART, 2002) para todo o conjunto de dados e encontra os mínimos e máximos locais a partir das distâncias alcançáveis calculadas.
- *Sparse modeling representative selection (SMRS)*: o método *SMRS* (ELHAMIFAR; SAPIRO; VIDAL, 2012) foi aplicado para todo o *dataset* sem nenhuma combinação;
- *K-Means* e *SMRS* (*k-means SMRS*): o conjunto é divido em k subgrupos com o algoritmo *K-Means*, para cada subgrupo é aplicado o algoritmo *SMRS* para obter as amostras mais significantes para cada subgrupo;
- *OPTICS* e *SMRS* (*OPTICS-SMRS*): utilizando uma metodologia similiar à anterior, porém utiliza o algoritmo *OPTICS* para criar os subgrupos;
- Kennard-Stone (*KenStone*): Amostragem utilizando o algoritmo de Kennard-Stone (*KenStone*) (KENNARD; STONE, 1969);
- Seleção aleatória: Seleciona aleatoriamente $p\%$ do conjunto;
- Seleção aleatória melhorada: Utiliza o algoritmo *K-Means* para criar subgrupos e seleciona aleatoriamente n_k elementos.

Foram utilizados quatro algoritmos semissupervisionados: dois baseados em grafos (funções harmonicas e grafo *anchor*), separações de menor densidade e múltiplos regressores baseados em suavidade. O trabalho conclui que, para algoritmos de aprendizado semissupervisionado, métodos de seleção baseados em densidade demonstraram resultados melhores que os tradicionais como a seleção aleatória. Além disso, algoritmos semissupervisionados baseados em grafos tiveram resultados superiores estatisticamente aos demais.

Os autores em ELHAMIFAR; SAPIRO; VIDAL (2012), já citados anteriormente em PARSAZAD; SABOORI; ALLAHYAR (2012), propõem encontrar uma amostra que descreve o conjunto assumindo que cada elemento pode ser expressado como uma combinação linear dos mais representativos, deste modo, achar os elementos representativos do conjunto é um problema de vetor de múltiplas medições esparsas. O método proposto foi comparado com outros métodos de seleção de elementos representativos em algoritmo de classificação, utilizando a amostra dada como representativa do método como treino. O método proposto chamado *Sparse modeling representative selection (SMRS)* conseguiu resultados significativos em três dos quatro algoritmos de classificação testados no conjunto de dados *USPS* (HULL, 1994) e em dois dos quatro testados no conjunto de dados *Extended YaleB* (LEE; HO; KRIEGMAN, 2005).

Já em PEIKARI et al. (2018) foi proposto um método de aprendizado semissupervisionado "Agrupar-então-Rotular" para classificação de imagens de patologia, o objetivo é encontrar regiões de alta densidade no conjunto de dados e usá-las para treinar um modelo confiável. Os autores discutem que os resultados indicam que apesar dos algoritmos de aprendizado semissupervisionado serem úteis para o problema em questão, é necessário identificar se a suposição de agrupamento são válidas para alguns projetos.

Métodos de aprendizado ativos selecionam os melhores elementos para serem rotulados COHN; ATLAS; LADNER (1994). Em TUR; HAKKANI-TÜR; SCHAPIRE (2005), dois métodos combinados com aprendizados ativo e semissupervisionado são propostos, ambos selecionam os rótulos dos elementos não rotulados por meio da confiança que o modelo ativo, treinado com dados já rotulados anteriormente, retorna sobre os rótulos classificados. O primeiro método utiliza os elementos rotulados com maior confiança pelo modelo ativo ao concatenar os dados não rotulado com os elementos rotulados e treinar um novo modelo com o conjunto concatenado, enquanto o segundo método utiliza a mesma metologia para rotulação porém ao invés de concatenar, acrescenta ao classificador ao mudar à função de perda até que ajuste-se ao modelo inicial e ao conjunto recém rotulado. O trabalho concluiu que para o reconhecimento de fala, foi possível utilizar menos da metade de elementos rotulados no conjunto enquanto que manteve a mesma performance.

Modelos de aprendizado semissupervisionado incremental geralmente selecionam os exemplos não rotulados com mais alta confiança de predição para o retreino do modelo. Porém resultados em ZHANG et al. (2005) demonstram que adicionar exemplos não

rotulados com baixa confiança obteve melhores resultados que adicionar exemplos rotulados com alta confiança na tarefa de reconhecimento de fala. Assim, o trabalho em ZHANG; RUDNICKY (2006) introduz um critério de seleção de exemplos baseado no potencial de contribuição para o treino do modelo, que é calculado da seguinte forma: o conjunto não rotulado é particionado em n subconjuntos como candidatos para ser selecionado, em seguida, uma função objetiva é usada para medir a capacidade de cada subconjunto em melhorar a acurácia do modelo, e o subconjunto que possuir a maior medida com os rótulos dados automaticamente são adicionados ao conjunto de dados de treino já existente. O autor conclui o trabalho demonstrando a efetividade do novo método de seleção de elemento comparados ao método atual, utilizando o *dataset* de referência *UCI*.

Outro trabalho próximo ao proposto neste projeto, ARAÚJO; ZHAO (2013a) propõem rotulação manual a partir de uma amostra gerada por métodos heurísticos baseados em métricas de centralidade de seleção de elemento para algoritmos semissupervisionados baseados em rede complexas. Foram utilizadas sete medidas de centralidade: *degree*, *neighboring degree variation*, *Page Rank*, coeficiente de agrupamento local, *neighboring clustering coefficient variation*, *local efficiency* e *local betweeness*. Além disso, os autores utilizam agregações das medidas feitas utilizando o algoritmo *PCA (Principal Component Analysis)* para identificar qual característica sumariza melhor cada tipo de rede complexa. Três algoritmos semissupervisionados baseados em redes complexas foram utilizados: *Gaussian Fields and Harmonic Functions (GFHF)*, *Local and Global Consistency (LGC)* e *Particle Competition and Cooperation (PCC)*. Para avaliar o resultado da propagação dos rótulos foi utilizado a métrica *Adjusted Rand Index (ARI)* (HUBERT; ARABIE, 1985), que varia de -1 até 1 e aumenta de acordo que o rótulo propagado é igual ao rótulo real do nó. O trabalho concluiu que nós altamente conectados (*hub*) não demonstraram ser elemento representativos em relação ao conjunto total, possuindo uma performance pior ou parecida que a seleção aleatória, a única exceção é quando o *hub* está conectado com nós de baixa conectividade. Em redes homogêneas elementos com alto coeficiente de agrupamento (*clustering*) são bons representantes do conjunto e, para redes heterogêneas, elementos com alta medida de *betweenness* são bons representante. Além disso, também conclui que com as agregações por PCA, o segundo componente principal (Z_2) exibiu resultados promissores.

Noutro trabalho, ARAUJO; ZHAO (2016) apresentam um método de construção de grafos que apresentou algumas propriedades como adaptabilidade à variação de densidade dos dados e pouca dependência de parâmetros escolhidos, além do método de construção, os autores propõem aplicar métricas de centralidade no grafo gerado para seleção dos elementos representativos para ser rotulados. As métricas de centralidade utilizados pelos autores: *degree*, índice de Katz, Page Rank, *Random Walk Closeness*, *Closeness* Harmônico, Coeficiente de agrupamento, *betweenness*, variação de grau hierárquico, variação de coeficiente de agrupamento hierárquico e variação de *betweenness* hierárquico. O estudo

conclui que há evidências que a representatividade de nós apresenta resultados para o aprendizado semissupervisionado e pode ser expandido em trabalhos futuros.

Em ARAÚJO; ZHAO (2013b), o trabalho explorou métricas de centralidade para seleção de nós representativos para rotulação com o objetivo de serem aplicados em algoritmos semissupervisionados baseados em redes complexas, de modo que os nós identificados apresentem perfis não homogêneos. O método proposto utiliza o índice de não-homogeneidade para cada nó i em $G(V, E)$, gerado a partir do conjunto no qual:

$$\lambda_i = (\varsigma_i - \langle\varsigma\rangle) \tag{4.1}$$

onde ς_i é o valor da métrica de centralidade de interesse para o nó i e $\langle\varsigma\rangle$ é a média dos valores da métrica de centralidade de interesse de todos os nós da rede, e quanto maior o valor de λ maior o índice. Os elementos selecionados para serem rotulados serão sempre os n nós que tiverem o maior índice, e as métricas de centralidade utilizadas foram: *degree*, eficiência local, coeficiente de agrupamento local, *betweenness* local, *Page Rank* e variação do coeficiente de agrupamento vizinho. Os autores concluem que o estudo parece indicar que a seleção de nós representativos para rotulação é essencial para a performance de métodos de aprendizado semissupervisionados baseados em redes. Para cada modelo testado, pelo menos duas estratégias performa melhor que a seleção aleatória.

A Tabela 4 sumariza os trabalhos relacionados aqui descritos. Destaca-se que os trabalhos mais semelhantes são de ARAÚJO; ZHAO (2013a) e ARAUJO; ZHAO (2016). Um diferencial, é que nesse projeto, além de medidas de centralidade analisou-se a distribuição de dados rotulados por comunidade.

Tabela 4 – Sumarização dos trabalhos

Trabalho	Objetivo
(PARSAZAD; SABOORI; ALLAHYAR, 2012)	Adaptar o algoritmo aiNet para seleção do subconjunto de dados a ser rotulado para algoritmos semissupervisionados
(PROTOPAPADAKIS; VOULODIMOS; DOULAMIS, 2018)	Explorar a efetividade de diferentes tipos de seleção de amostras, propondo soluções em uma variedade de combinações de algoritmos de amostragem
(ARAÚJO; ZHAO, 2013a)	Adotar critérios heurísticos baseados em métricas de centralidade para seleção de amostras para aprendizado semissupervisionado baseado em redes complexas
(ELHAMIFAR; SAPIRO; VIDAL, 2012)	Utilizar de modelagem dispersa para encontrar exemplos que descrevam todos o conjunto de dados de forma eficiente e com esse conjunto classificar exemplos de sumarização de vídeos e imagens
(PEIKARI et al., 2018)	Aplicar o método de aprendizado semissupervisionado Agrupar-então-Rotular em classificação de patologias em imagens
(ARAUJO; ZHAO, 2016)	Propõem um método adaptativo de construção de grafo que considera heterogeneidade de interação local chamado *AdaRadius* e um método heurístico para seleção de nós representativos a serem rotulados
(ARAÚJO; ZHAO, 2013b)	O trabalho propõe utilizar métricas de centralidade de redes complexas para selecionar nós representativos para rotulação manual que têm a possibilidade de melhorar a performance de algoritmos de aprendizado semissupervisionado baseados em redes complexas
(TUR; HAKKANI-TÜR; SCHAPIRE, 2005)	Combinar aprendizado ativo e aprendizado semissupervisionado para minimizar o esforço em rotular dados de reconhecimento de fala. Selecionado elementos para serem rotulados pelo aprendizado ativo para depois utilizar o aprendizado semissupervisionado.
(ZHANG; RUDNICKY, 2006)	O trabalho propõe uma nova medida baseada na contribuição do exemplo no treino, em vez da confiança do exemplo na previsão.

5 Material e métodos

Esse trabalho visa analisar a influência dos elementos rotulados no desempenho de algoritmos semissupersivionados baseados em grafos, para isso empregamos os seguintes passos (ilustrados na Figura 1):

1. Seleção de *datasets*: utilizamos um *benchmark* (CHAPELLE; SCHOLKOPF; ZIEN, 2009) proposto especialmente para o aprendizado semissupervisionado. Esses dados são descritos em mais detalhes na seção a seguir 5.1.

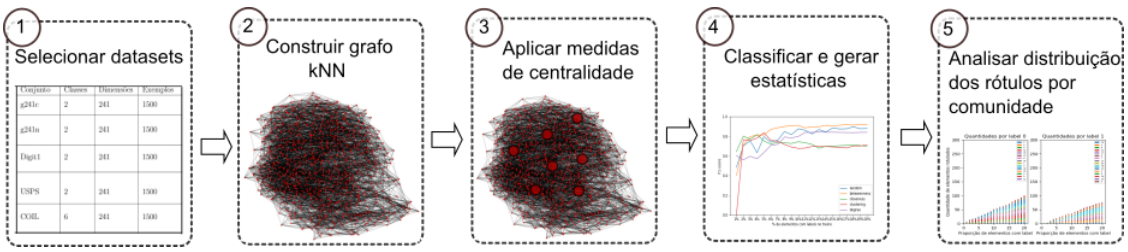
2. Geração de grafos: construímos grafos a partir do conjunto de dados por meio do algoritmo *k-Nearest Neighbors* (kNN) utilizando as métricas de distância Euclidiana (Minkowski com $p = 2$), com $k = 5$. Já que valores menores de k são mais apropriados para evitar grafos muito densos e consequentemente ruído na propagação dos rótulos (BERTON, 2016);

3. Seleção de amostras rotuladas: aplicamos medidas de centralidade para encontrar os exemplos mais representativos das classes. As medidas utilizadas foram: *degree*, *closeness*, *betweeness* e *clustering*. Foram selecionados $x\%$ de elementos com os maiores valores de centralidade, sendo que $1 \leq x \leq 20$. Também fizemos uma seleção aleatória de elementos no conjunto de dados inicial como *baseline*;

4. Classificação: utilizamos os conjuntos de dados para treinar algoritmos semisupervisionados e classificar os dados não-rotulados. Foram usados os algoritmos *Local and Global Consistency* (LGC) e *Harmonic Functions* (HMN). Para a avaliação foi empregada a métrica de *F1-score* nos resultados dos classificadores.

5. Detecção de comunidades: por fim, o algoritmo de detecção de comunidades *Louvain* foi aplicado nos grafos, a fim de contabilizar a quantidade de comunidades geradas e a quantidade de elementos rotulados em cada comunidade.

Figura 1 – Passos para realização do trabalho



Além disso, foram executados alguns processamentos nos grafos, a fim de melhorar a aplicação dos classificadores. Primeiro, como podem haver grafos com subgrafos não

conexos, o maior subgrafo conexo foi utilizado para a seleção de rótulos e aplicação dos algoritmos de classificação. Foram realizados dois experimentos onde 1) o grafo foi mantido inalterado; 2) nós vizinhos com rótulos diferentes foram desconectados.

5.1 Conjuntos de Dados

Os conjuntos de dados explorados são *datasets* tradicionalmente empregados no aprendizado semissupervisionado proposto por (CHAPELLE; SCHÖLKOPF; ZIEN, 2006) são resumidos na Tabela 5.

Na Figura 2, é possível observar a distribuição dos dados usando o algoritmo de redução de dimensionalidade PCA, ou *Principal Component Analysis* (TIPPING; BISHOP, 1999), para todos os conjuntos. Nas Figuras 2 (b) e (c), os conjuntos *Digit1* e *g241c*, respectivamente, possuem uma separação com pouca sobreposição entre os rótulos 0 e 1. Já o *dataset g241n* na Figura 2 (d), possui dois agrupamentos porém cada grupo possui mistura entre os rótulos. Enfim, os conjuntos *COIL*, na Figura 2 (a), e *USPS*, na Figura 2 (e), não possuem uma separação muito clara entre os rótulos, com bastante sobreposição.

5.2 Bibliotecas Utilizadas

As seguintes bibliotecas foram utilizadas no desenvolvimento deste trabalho:

- **NumPy:** É uma biblioteca da linguagem Python que permite trabalhar com arranjos, vetores e matrizes de N dimensões. Provê diversas funções e operações sofisticadas (WALT; COLBERT; VAROQUAUX, 2011).
- **Matplotlib:** Biblioteca para gerar gráficos 2D em Python (HUNTER, 2007).
- **Seaborn:** Biblioteca de visualização de dados que cria uma interface de alto nível para gerar gráficos e informativos baseado no Matplotlib (WASKOM et al., 2014).
- **Pandas:** Biblioteca para Python que provê estruturas de dados e ferramentas para analisá-las, baseada na biblioteca do NumPy (MCKINNEY et al., 2010).
- **Scikit-learn:** Biblioteca que provê métodos de mineração de dados, aprendizado de máquina e análise de dados, também baseada na biblioteca do NumPy (PEDREGOSA et al., 2011).
- **NetworkX:** Biblioteca de análise e exploração de algoritmos de redes complexas em Python (HAGBERG; SCHULT; SWART, 2008).
- **Label Propagation:** Biblioteca com implementações de algoritmos de propagação de rótulos baseados em grafos (YAMAGUCHI, 2020).

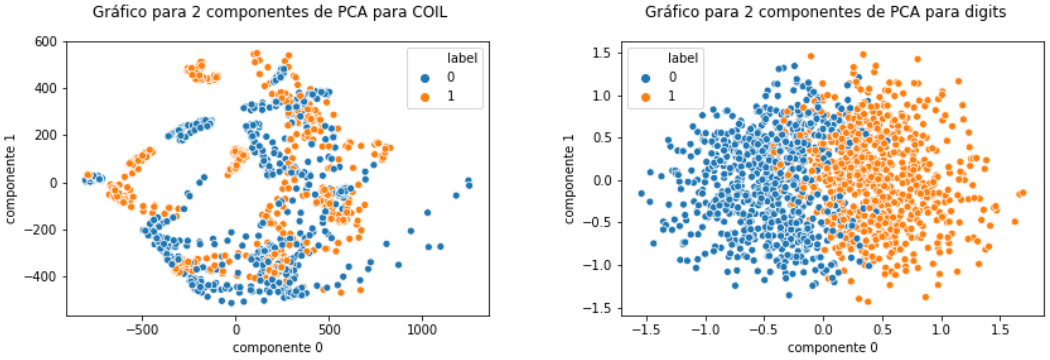
Tabela 5 – Propriedades dos conjuntos de dados

Conjunto	Classes	Dimensões	Exemplos	Características
g241c	2	241	1500	Pontos gerados artificialmente respeitando a suposição de agrupamento.
g241n	2	241	1500	Pontos gerados artificialmente não respeitando a suposição de agrupamento e sem estrutura de *manifold*.
Digit1	2	241	1500	Imagens do dígito 1 geradas artificialmente para possuir uma baixa dimensionalidade de *manifold* dentro de um espaço de alta dimensionalidade e para não possuir totalmente uma estrutura de agrupamento. Nas discussões, o conjunto foi referenciado como *digits* ou *Digit1*
USPS	2	241	1500	Derivado do conjunto de dados com o mesmo nome (USPS), foram retirados 150 imagens de cada dígito, e os dígitos 2 e 5 foram atribuídos o valor de classe +1 enquanto os outros, -1.
COIL2	2	241	1500	O conjunto de dados é derivado do conjunto COIL-100, que é um *dataset* de imagens de objetos em cores tiradas em diferentes ângulos com uma resolução de 128 $\times$ 128 pixels (NENE; NAYAR; MURASE, 1996). A qualidade das imagens foi diminuída para 16 $\times$ 16 fazendo uma média nos blocos de pixels de 8 $\times$ 8. Dos 100 objetos foram selecionados 24 que foram separados em 6 classes de 4 objetos cada. Foram descartados 38 imagens de cada classe e aplicado o algoritmo de obscurecimento de imagem em (CHAPELLE; SCHÖLKOPF; ZIEN, 2006) com $\sigma = 2$. Entretanto no nosso projeto foi utilizado a versão binária do conjunto de dados, e, nas discussões, o conjunto foi referenciado somente como COIL.

Fonte: (CHAPELLE; SCHÖLKOPF; ZIEN, 2006)

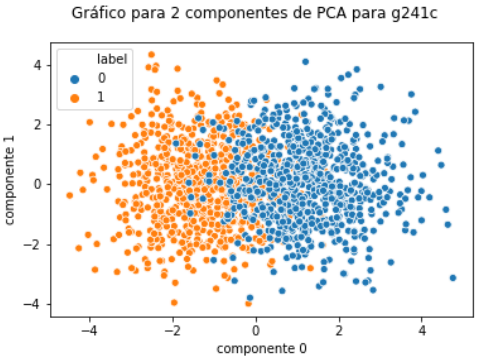
- **python-louvain:** Biblioteca para análises de comunidade em redes complexas e grafos com o algoritmo de detecção de comunidades de Louvain (AYNAUD, 2020).

Figura 2 – Distribuição dos dados usando PCA: (a) *COIL* (b) *Digit1* (c) *g241c* (d) *g241n* (e) *USPS*

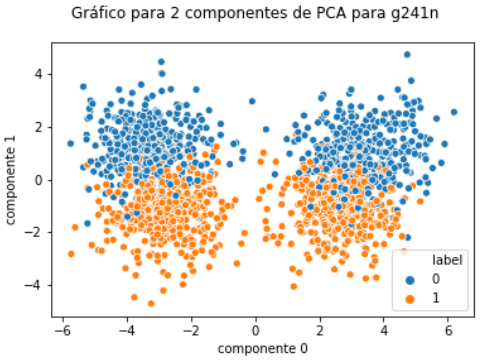


(a)

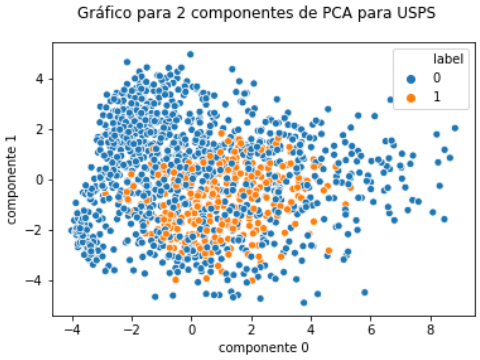
(b)



(c)



(d)



(e)

6 Resultados

Nesse capítulo são descritos os principais resultados obtidos. Na seção 6.1 é apresentado o experimento sem desconectar arestas com rótulos diferentes e na seção 6.2 desconectando. Dentro de cada seção apresenta-se um resumo e os resultados completos. variando porcentagem de elementos rotulados e análise comparativa da quantidade de rótulos por comunidades.

6.1 Experimento 1: sem desconectar arestas com rótulos diferentes

6.1.1 Sumarização dos Resultados

As Tabelas 6 e 7 sumarizam os resultados comparando o maior *F1-score* alcançado por cada medida de centralidade e pela seleção aleatória de elementos rotulados. Em geral, com o uso das medidas é possível obter resultados melhor que a seleção aleatória. Na média, a medida *betweenness* obtém melhor colocação e *closeness* a pior. E os resultados de classificação não tiveram muita influência pelo classificador empregado, já que tanto HMN quanto LGC obtiveram acurácias semelhantes.

Tabela 6 – Melhores resultados de cada seleção (*F1-score*) com HMN

Conjunto de dados	*degree*	*clustering*	*closenness*	*betweenness*	aleatório
Digit1	0,981	0,983	0,984	**0,985**	0,982
USPS	0,840	0,836	0,802	**0,912**	0,887
COIL	0,488	0,530	0,463	0,513	**0,533**
g241n	**0,724**	0,685	0,712	0,722	0,695
g241c	0,665	**0,672**	0,671	0,667	0,664
média	0,7398	0,7412	0,7264	**0,7598**	0,7522

Tabela 7 – Melhores resultados de cada seleção (*F1-score*) com LGC

Conjunto de dados	*degree*	*clustering*	*closenness*	*betweenness*	aleatório
Digit1	0,981	0,984	0,983	**0,985**	0,982
USPS	0,676	0,819	0,768	**0,881**	0,789
COIL	0,486	**0,530**	0,463	0,502	**0,533**
g241n	**0,742**	0,693	0,686	0,687	0,690
g241c	0,659	0,649	0,656	**0,665**	0,657
média	0,7088	0,735	0,7112	**0,744**	0,7302

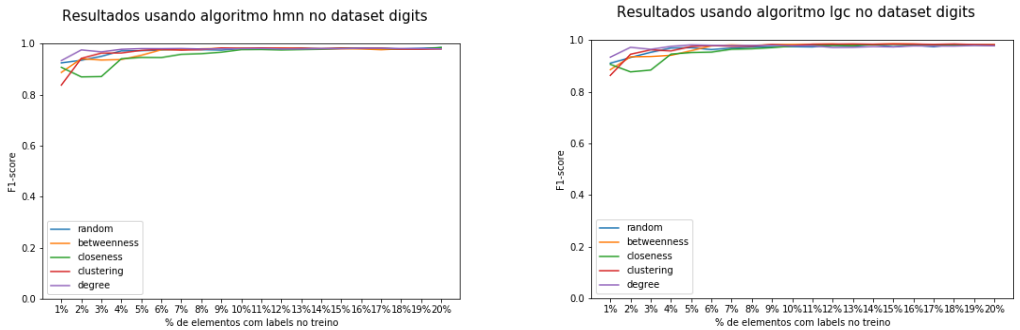
6.1.2 Resultados completos

Nesse experimento, os grafos para cada *dataset* foram gerados pelo algoritmo *k-Nearest-Neighbors*, com $k = 5$. Selecionamos o maior subgrafo conexo, e o grafo final será nomeado como G. Para cada grafo gerado, as métricas de centralidade foram calculadas, os $x\%$ elementos com os maiores valores de centralidade serão rotulados, sendo que $1 \leqslant x \leqslant 20$. Em seguida, o grafo G foi passado para os algoritmos de classificação LGC e HMN. Os rótulos dos nós propagados foram utilizados para o cálculo da métrica *F1-score*. Por fim, o algoritmo Louvain foi empregado para detectar o número de comunidades geradas em cada grafo e uma análise foi feita com relação a quantidade de elementos rotulados por comunidade.

6.1.2.1 *Dataset Digit1*

O *dataset Digit1* se mostrou mais fácil de classificar com resultados próximo de 1.0 para todas as medidas como pode ser observado em Figura 3. Considerando até 3% de dados rotulados, a medida *degree* obteve resultados melhores que as demais, porém, com o aumento de rótulos o *F1-Score* ficou próximo do *clustering* e da seleção aleatória. Já os piores resultados foram obtidos pelas medidas *betweenness* e *closeness* (considerando até 10% dos elementos rotulados). A partir de 10% de rótulos todas as medidas tiveram desempenho semelhante. O desempenho do HMN e LGC ficaram bastante próximos.

Figura 3 – *F1-Score* por % de elementos rotulados: a) HMN e b) LGC.



(a) (b)

A Figura 4 mostra o total de elementos rotulados de cada classe (0/1) selecionados pelas diferentes medidas de centralidade, variando-se a porcentagem de rótulos de 1 até 20 %. Note que para poucos dados rotulados (até 5%) a distribuição de rótulos por classe varia bastante, especialmente para *clustering* e *closeness*. No geral, a quantidade de rótulo 0 e 1 é equivalente para todas as medidas.

É possível observar que a distribuição de elementos rotulados entre as comunidades parece afetar o desempenho da classificação. A medida *degree* foi melhor desde o início, e tem uma distribuição de dados rotulados mais balanceada entre as comunidades, o que

Figura 4 – Quantidade de comunidades com elementos rotulados selecionados pelas diferentes medidas de centralidade aplicadas no *dataset Digit1*.

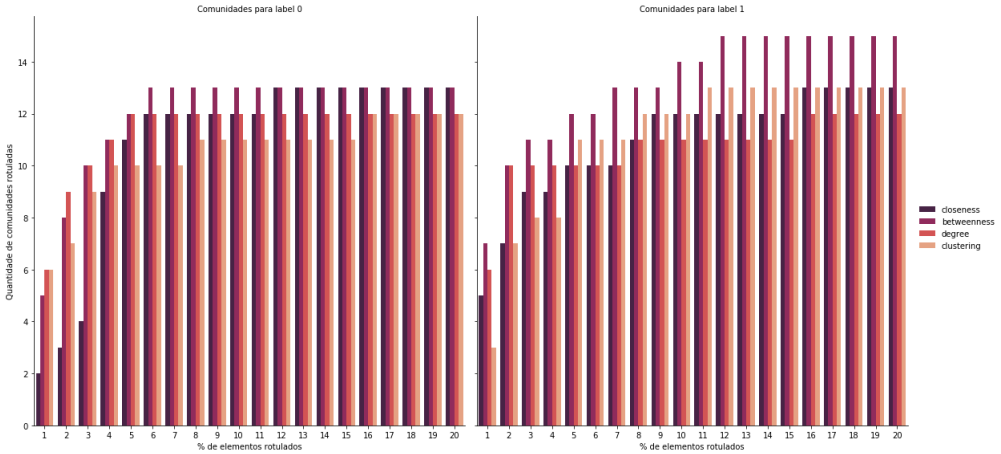
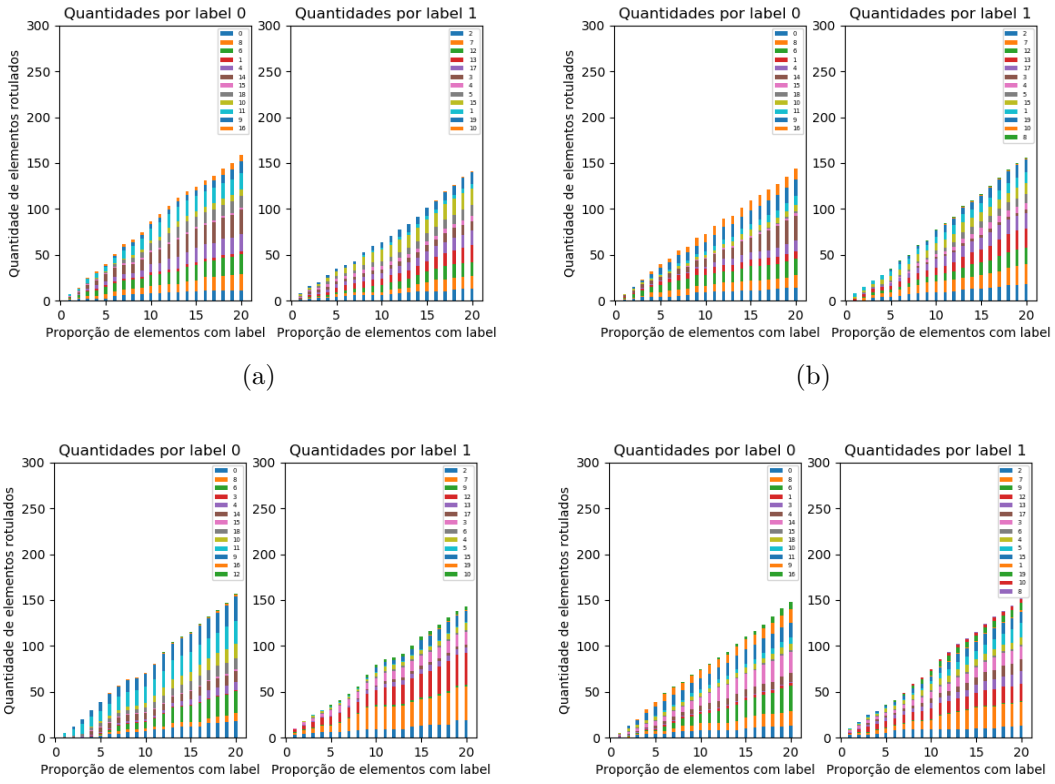


Figura 5 – Distribuição de rótulos (0/1) por comunidades (representadas por cores) para as medidas de centralidade: (a) *degree* (b) *clustering* (c) *closeness* (d) *betweenness*



(a) (b)

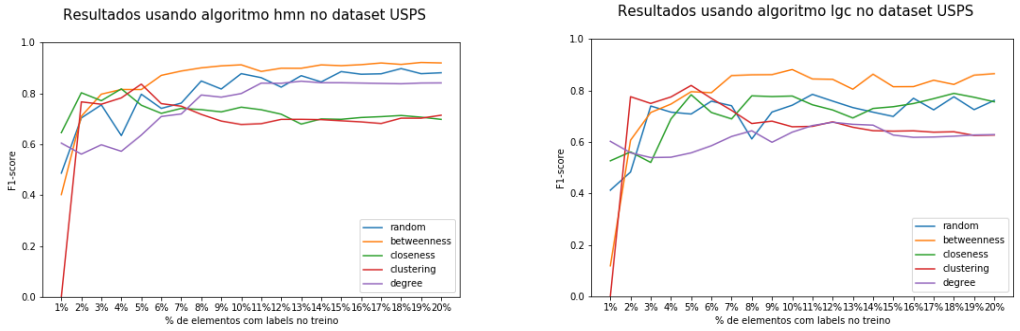
(c) (d)

pode ser visualizado na Figura 5 (a). Já a medida *closeness*, não teve uma distribuição de dados rotulados balanceada entre as comunidades (Figura 5 (c)) e obteve pior desempenho na classificação. Note que o rótulo 0 ficou mais presente nas comunidades azul claro e escuro enquanto o rótulo 1 ficou mais presente nas comunidades vermelho e laranja. No geral, todas as medidas tiveram uma distribuição equivalente de rótulos entre as comunidades e por isso, provavelmente, obtiveram bom desempenho em Digit1.

6.1.2.2 *Dataset USPS*

Conforme pode ser visto na Figura 6, o comportamento do *F1-Score* variou bastante de acordo com a medida de centralidade e proporção de elementos selecionados. *Betweenness* foi consistente e melhorou a medida que a proporção de rótulos aumentava. O *degree* também manteve uma tendência de melhoria, apesar de ter obtido continuamente um *score* pior que o *betweeness* e da seleção aleatória. Tanto a medida *clustering* quando *closeness* tiveram um *F1-score* próximo ao *betweenness*, porém depois de 5% de proporção de elementos rotulados, o *score* ficou pior que as demais. HMN e LGC tiveram desempenho semelhante, porém, em LGC a medida *clustering* se manteve a pior posição em todas as porcentagens de rótulos consideradas.

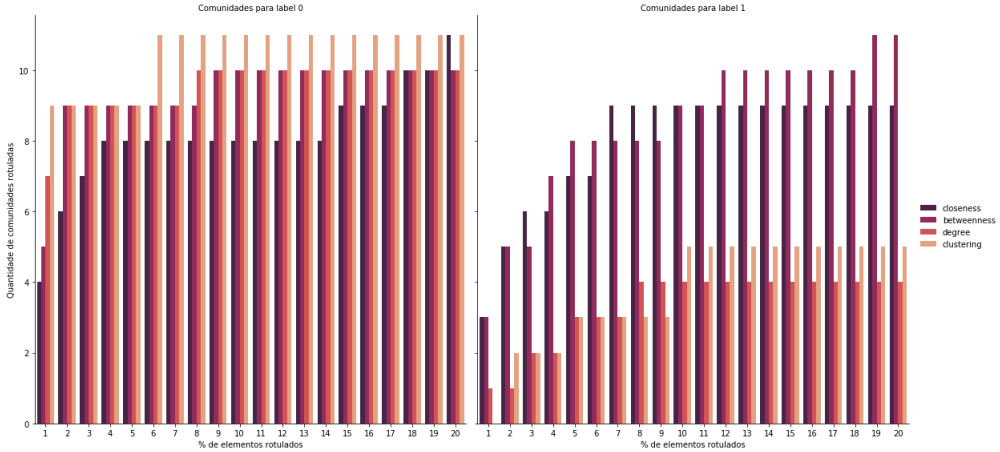
Figura 6 – *F1-Score* por % de elementos rotulados: a) HMN e b) LGC.



(a) (b)

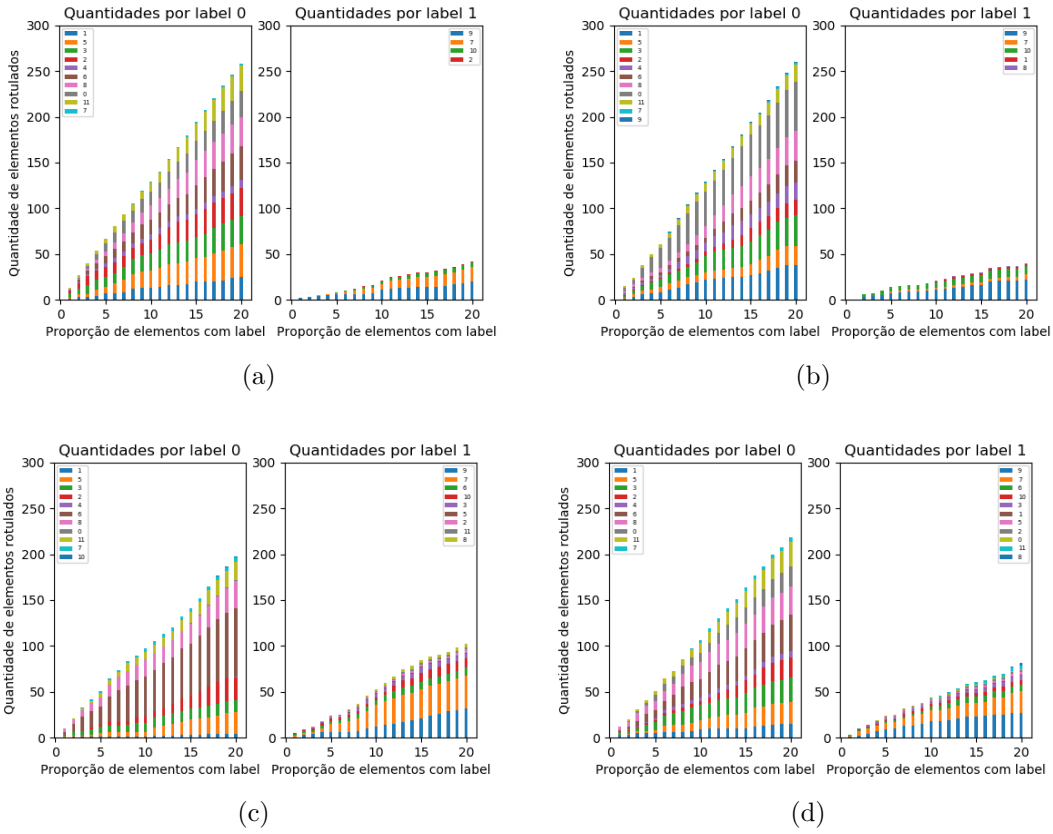
Como pode ser observado na Figura 7 a quantidade de elementos rotulados ficou mais proporcional para as medidas *closeness* e *betweeness*. Porém, apenas *betweeness* obteve bom desempenho na classificação. Já em *clustering* e *degree* a distribuição de elementos rotulados ficou desbalanceada (ou seja, com mais elementos da classe 0).

Figura 7 – Quantidade de elementos rotulados selecionados pelas diferentes medidas de centralidade aplicadas no *dataset USPS*.



Com relação a distribuição de elementos rotulados por comunidades para o *dataset* USPS nota-se que grande parte das medidas possuem muitas comunidades com o rótulo 0 e poucas comunidades com o rótulo 1. A medida *betweeness* é a única que possui várias comunidades com o rótulo 1 e provavelmente por isso obteve melhor desempenho (ver Figura 8 (d)).

Figura 8 – Distribuição de rótulos (0/1) por comunidades (representadas por cores) para as medidas de centralidade: (a) *degree* (b) *clustering* (c) *closeness* (d) *betweenness*

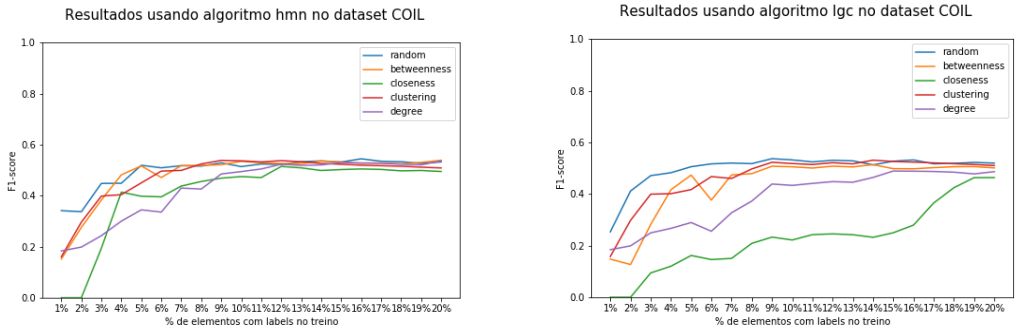


6.1.2.3 *Dataset COIL*

A Figura 9 apresenta o resultado do *F1-score* para o *dataset* COIL. No geral, o desempenho nesses dados foi pior que nos demais conjuntos de dados analisados. As medidas *degree* e *closeness* performaram pior que as demais tanto no HMN quanto no LGC.

Nota-se na Figura 10 que os dados rotulados ficaram bastante desproporcionais em todas as medidas de centralidade. Essa desproporção também é notada na Figura 11, onde a quantidade de elementos do rótulo 1 sempre foi inferior ao rótulo 0. Provavelmente devido a esse desbalanceamento na seleção de rótulos, esse *dataset* teve uma acurácia mais baixa. Nota-se que *closeness* e *betweenness* (Figura 11c e d), apesar da desproporção

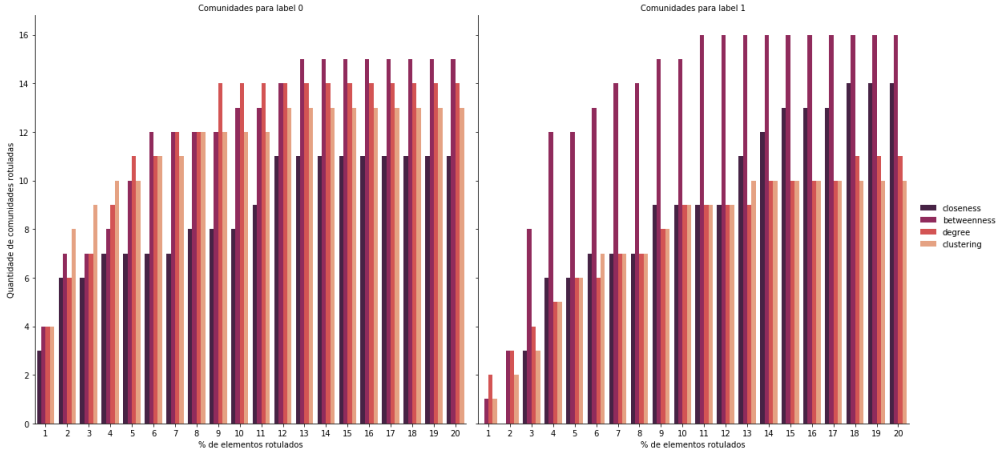
Figura 9 – *F1-Score* por % de elementos rotulados: a) HMN e b) LGC.



(a) (b)

der rótulos, tiveram a presença de ambos os rótulos em várias comunidades, e com isso alcançaram um desempenho maior na classificação comparado ao *degree* e *clustering*.

Figura 10 – Quantidade de elementos rotulados selecionados pelas diferentes medidas de centralidade aplicadas no *dataset COIL*.



6.1.2.4 *Dataset g241c*

Em geral, os experimentos no *dataset g241c*, não alcançaram alto desempenho. Contudo, as medidas de centralidade foram melhores em quase todos os casos do que a seleção aleatória no classificador HMN. Apenas *clustering* teve um comportamento anômalo com 9% de elementos rotulados, onde a acurácia decaiu. No classificador LGC teve bastante oscilação nos resultados, *closeness* teve o pior desempenho e *clustering* também teve um comportamento anômalo com 9% de elementos rotulados (ver Figura 12).

Na Figura 13 observa-se que a proporção de elementos rotulados é similar para todas as medidas de centralidade. Apenas para seleção menor que 5% existe desbalanceamento entre as classes.

A Figura 14 mostra que a distribuição de elementos rotulados por comunidade também é semelhante entre todas as medidas. Provavelmente, por isso todas obtiveram

Figura 11 – Distribuição de rótulos (0/1) por comunidades (representadas por cores)para as medidas de centralidade: (a) *degree* (b) *clustering* (c) *closeness* (d) *betweenness*.

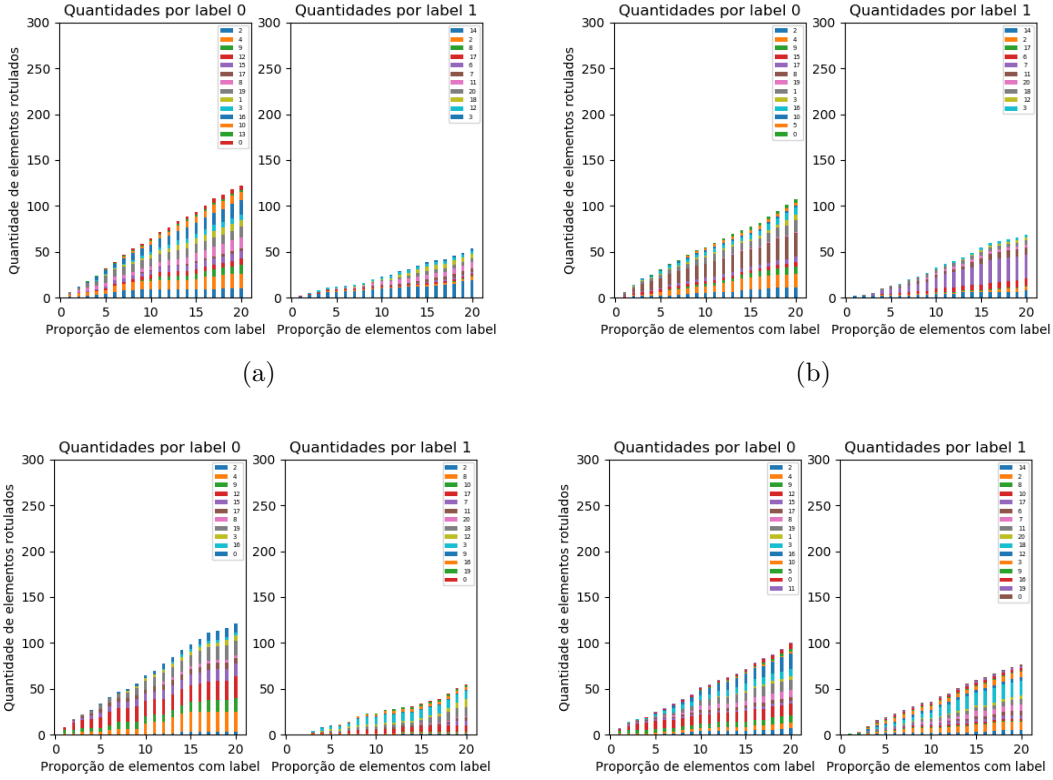
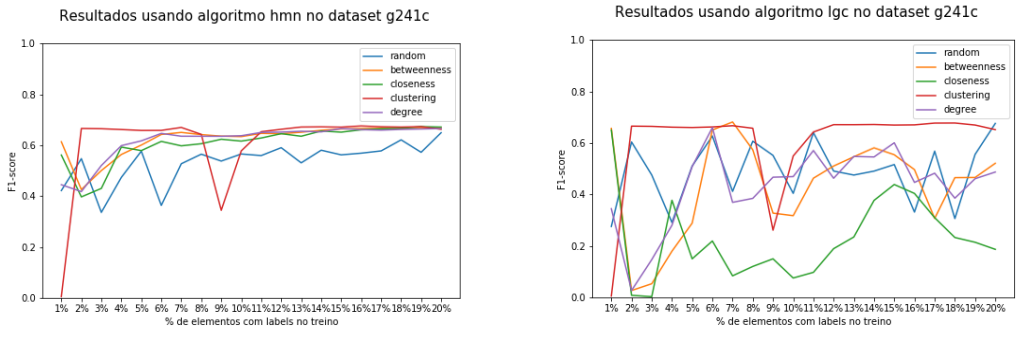


Figura 12 – *F1-Score* por % de elementos rotulados: a) HMN e b) LGC.



performance semelhante na classificação. Contudo, não conseguimos identificar o motivo do *dataset g241c* obter um *F1-score* mais baixo.

Os *datasets g241c e g241n* foram gerados artificialmente. O *g241n* apresentado a seguir, não tem estrutura de comunidades nem *manifold*. Provavelmente, por esse motivo, esses *datasets* não apresentaram um padrão entre a distribuição dos rótulos e a classificação.

Figura 13 – Quantidade de elementos rotulados selecionados pelas diferentes medidas de centralidade aplicadas no *dataset g241c*

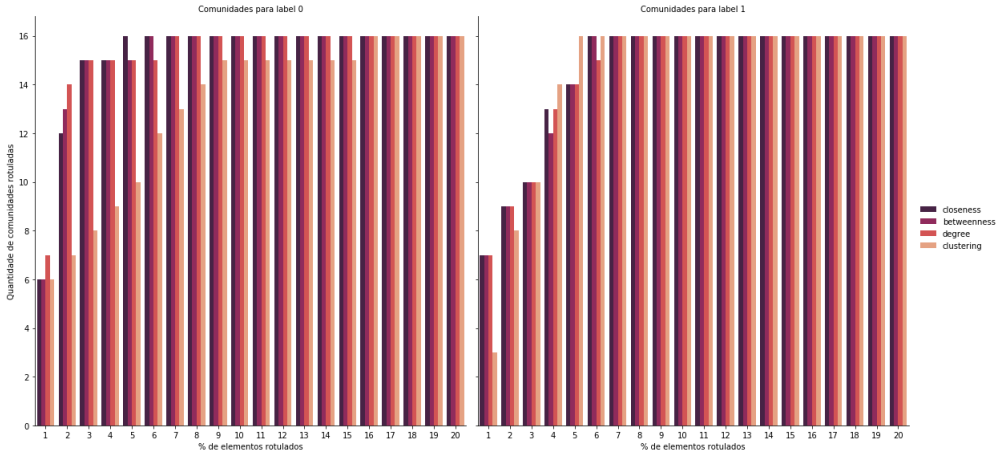
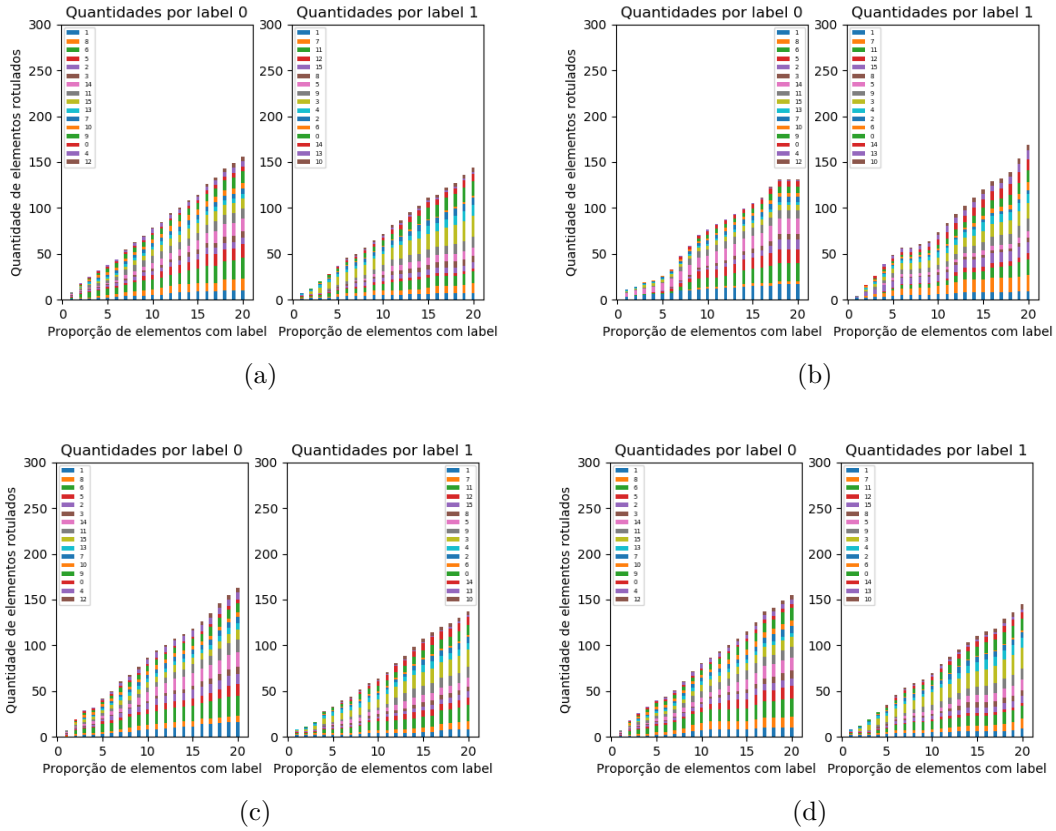


Figura 14 – Distribuição de rótulos (0/1) por comunidades (representadas por cores) para as medidas de centralidade: (a) *degree* (b) *clustering* (c) *closeness* (d) *betweenness*



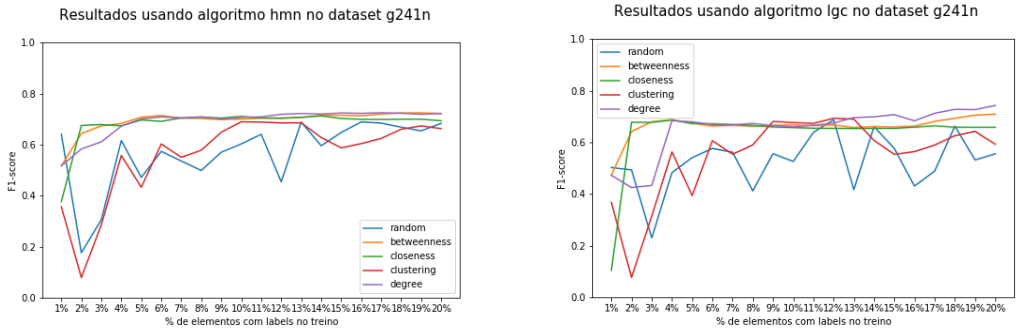
(a) (b)

(c) (d)

6.1.2.5 *Dataset g241n*

Os resultados no *dataset g241n* são próximos aos do *dataset g241c* (ver Figura 15). Todas as medidas tiveram desempenho semelhante, com exceção do *clustering*, o qual obteve desempenho inferior e próximo a seleção aleatória tanto no HMN como no LGC.

Figura 15 – *F1-Score* por % de elementos rotulados: a) HMN e b) LGC.



(a) (b)

A Figura 16 mostra que a distribuição de elementos rotulados foi semelhante para todas as medidas. Com menos de 5% de dados rotulados *clustering* apresentou mais elementos da classe 0.

Esse foi o único conjunto de dados com poucas comunidades, apenas três, sendo duas delas, bem mais predominantes que as demais, conforme mostrado na Figura 17. A distribuição de elementos rotulados entre as duas comunidades foi semelhante para todas as medidas.

Figura 16 – Quantidade de elementos rotulados selecionados pelas diferentes medidas de centralidade aplicadas no *dataset g241n*

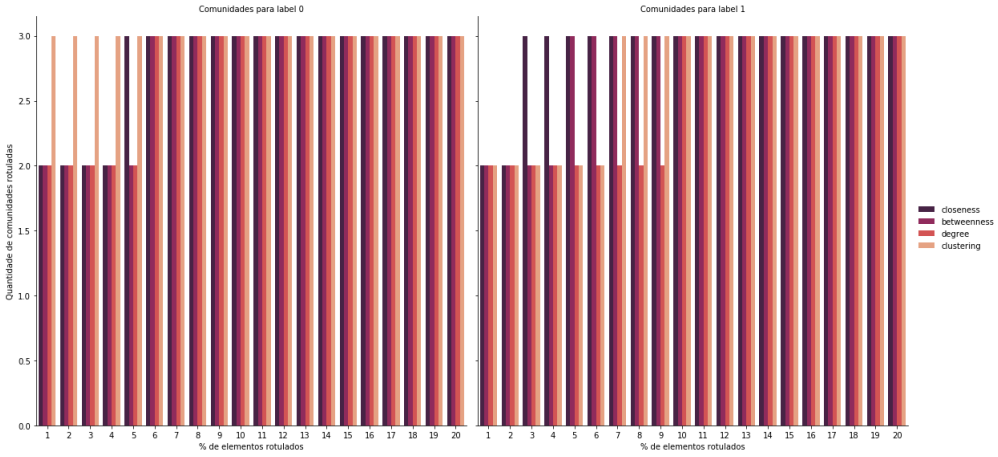
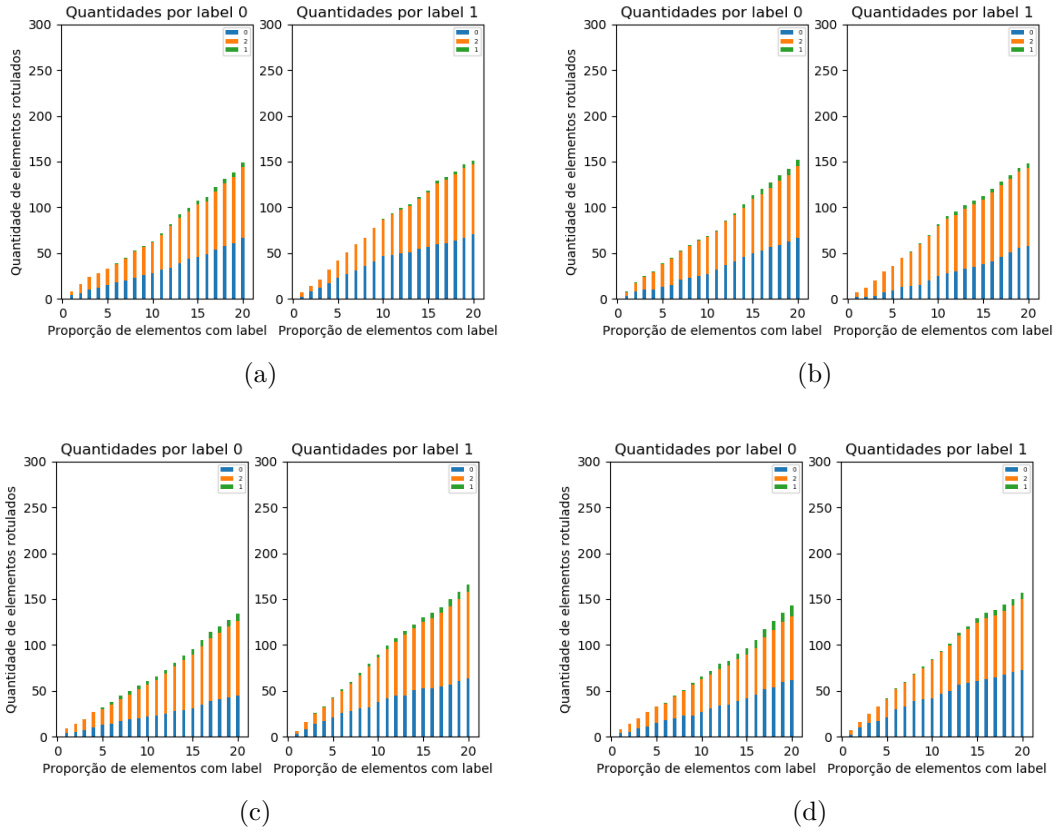


Figura 17 – Distribuição de rótulos (0/1) por comunidades (representadas por cores) para as medidas de centralidade: (a) *degree* (b) *clustering* (c) *closeness* (d) *betweenness*



6.2 Experimento 2: desconectando arestas com rótulos diferentes

6.2.1 Sumarização dos Resultados

Os resultados apresentados nas Tabelas 8 e 9, são os que possuíram maior *F1-score* para cada seleção. Na média, os resultados não diferiram muito dos resultados do experimento 1 nas Tabelas 6 e 7. Além disso, em ambos os experimentos a métrica *betweenness* teve melhor resultado em média.

Tabela 8 – Melhores resultados de cada seleção (*F1-score*) com HMN

Conjunto de dados	*degree*	*clustering*	*closenness*	*betweenness*	aleatório
Digit1	0,982	0,983	**0,984**	**0,984**	0,982
USPS	0,847	0,836	0,817	**0,921**	0,882
COIL	0,535	**0,538**	0,514	0,535	0,536
g241n	**0,724**	0,685	0,712	**0,724**	0,685
g241c	0,665	**0,675**	0,673	0,667	0,632
média	0,7396	0,7434	0,7400	**0,7662**	0,7434

Tabela 9 – Melhores resultados de cada seleção (*F1-score*) com LGC

Conjunto de dados	*degree*	*clustering*	*closenness*	*betweenness*	aleatório
Digit1	0,981	**0,986**	0,983	0,984	0,982
USPS	0,679	0,814	0,823	**0,913**	0,792
COIL	0,503	**0,531**	0,489	0,502	**0,533**
g241n	**0,739**	0,693	0,686	0,719	0,671
g241c	0,651	**0,677**	0,648	0,665	0,66
média	0,7106	0,7402	0,7258	**0,7566**	0,7276

6.2.2 Resultados completos

Esse experimento seguiu a mesma metodologia que o anterior para construir os grafos e selecionar os elementos, porém, depois da seleção de elementos, nós conectados que possuíam rótulos diferentes foram desconectados. Em seguida, os maiores subgrafos conexos foram selecionados, passados para os algoritmos de classificação LGC e HMN, e os rótulos propagados foram utilizados para o cálculo da métrica *F1-score*.

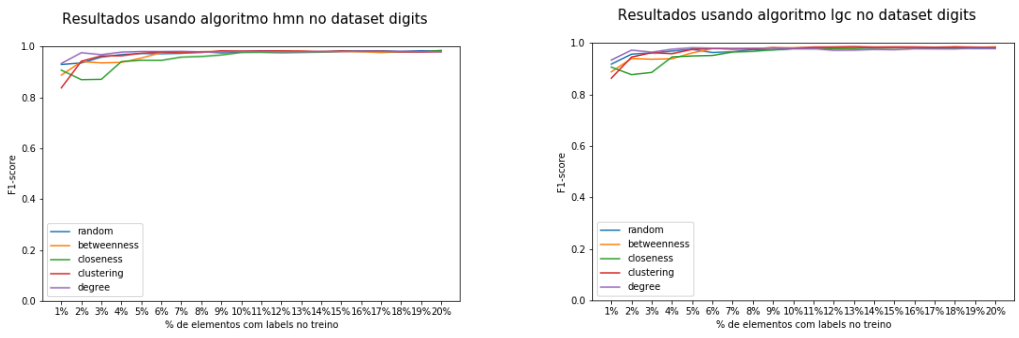
Nesse experimento, não foi possível aplicar o algoritmo de detecção de comunidades Louvain, já que a cada $x\%$ de elementos rotulados o grafo sofrerá alterações por conta dos nós desconectados e não necessariamente teríamos o mesmo grafo em todos os experimentos.

6.2.2.1 *Dataset Digit1*

Neste segundo experimento com o conjunto *Digit1*, não houveram alterações significativas ao comparamos os resultados dos dois algoritmos na Figura 18. Quando comparado com os resultados do mesmo conjunto de dados no Experimento 1 (presente na Figura 3), pode ser observado que os resultados foram semelhantes.

Portanto, os resultados indicam que, para este conjunto, não há melhorias em desconectar nós com rótulos diferentes, valendo destacar que o primeiro experimento com este conjunto já obteve resultados positivos comparando aos demais conjuntos.

Figura 18 – *F1-Score* por % de elementos rotulados: a) HMN e b) LGC.



(a) (b)

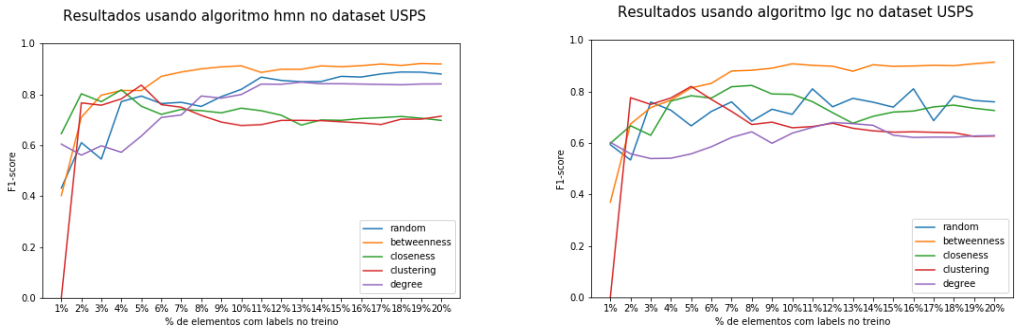
6.2.2.2 *Dataset USPS*

Diferente dos resultados com o *dataset Digit1*, houveram alterações nas medidas entre os algoritmos na Figura 19. Comparando os dois gráficos (ver Figura 19 (a) e (b)), observam-se duas diferenças notáveis:

- o resultado da seleção com *degree* no LGC teve um desempenho pior que a mesma seleção no HMN;
- o resultado do *closeness* no LGC desempenhou melhor que no HMN.

Ao compararmos os resultados desse experimento com os resultados do experimento anterior (ver Figura 6), percebemos que, em geral, os resultados se mantiveram similares com a exceção da seleção com *betweenness*. Essa seleção desempenhou melhor no algoritmo da LGC no experimento atual, porém em ambos tiveram os melhores resultados ao serem comparadas com as demais seleções de cada algoritmo no experimento.

Figura 19 – *F1-Score* por % de elementos rotulados: a) HMN e b) LGC.



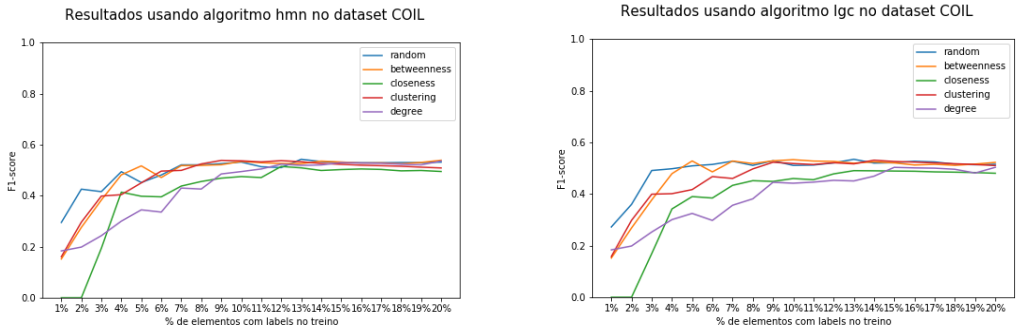
(a) (b)

6.2.2.3 *Dataset COIL*

Observando os resultados do experimento 2 com o conjunto de dados COIL (ver Figura 20), chegamos a conclusão que, com a separação de nós com rótulos diferentes, não há diferenças em desempenho quando comparamos o HMN e o LGC. Porém, ao compararmos com os resultados do experimento 1 (ver Figura 9), os resultados do LGC com *closeness* no experimento 2 tiveram um ganho de desempenho com tamanho da amostra entre 3% e 17%.

Mesmo com o ganho de desempenho da seleção com *closeness*, os resultados continuam piores que a seleção aleatória e somente *betweenness* e *clustering* possuíram desempenho similares.

Figura 20 – *F1-Score* por % de elementos rotulados: a) HMN e b) LGC.

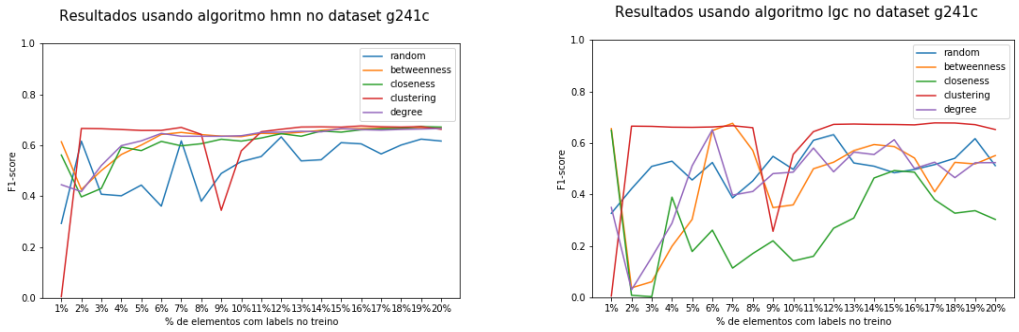


(a) (b)

6.2.2.4 *Dataset g241c*

Os resultados do experimento 2 com o conjunto de dados *g241c* (ver Figura 21) e os resultados do experimento 1 com o mesmo *dataset* (ver Figura 12) foram similares, inclusive mantendo a mesma anomalia no *clustering* com 9% de elementos rotulados.

Figura 21 – *F1-Score* por % de elementos rotulados: a) HMN e b) LGC.



(a) (b)

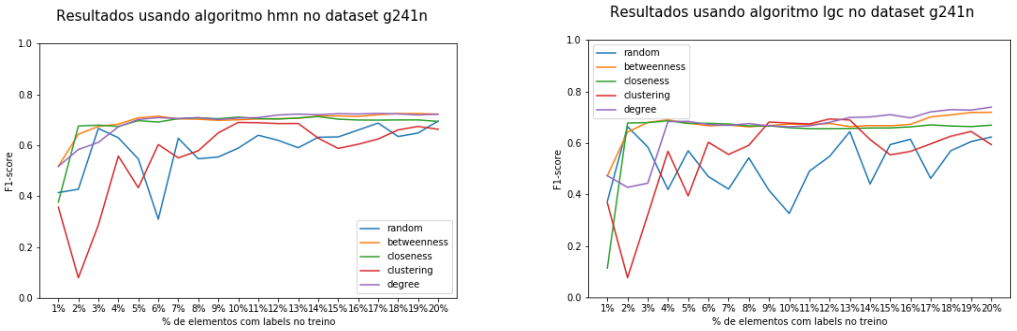
6.2.2.5 *Dataset g241n*

Analisando os resultados do experimento 2 com o *dataset g241n* (ver Figura 22), é possível observar que as seleções pelas métricas de centralidade tiveram poucas mudanças nos resultados dos algoritmos. As únicas diferenças sendo que as métricas *degree* e *betweenness* tiveram um desempenho melhor após a proporção de amostras rotuladas de 12%, e a métrica *clustering* desempenhando melhor entre as proporções de amostras rotuladas de 9% e 13%.

Porém, em contraste com as seleções por métricas de centralidade, a seleção aleatória teve resultados mais consistentes com o algoritmo HMN, já que as medidas tiveram menos variações ao longo das proporções de elementos rotulados. Ao compararmos os resultados

desse experimento (Figura 22) com os resultados do experimento 1 (Figura 15), percebemos que não houveram mudanças significativas.

Figura 22 – *F1-Score* por % de elementos rotulados: a) HMN e b) LGC.



(a) (b)

7 Conclusão

A seguir são descritas as principais contribuições e limitações do trabalho realizado.

7.1 Contribuições

Nesse trabalho foi realizado experimentos explorando a influência de elementos rotulados na classificação semissupervisionada e obtivemos os seguintes resultados:

1) Observamos que, em geral, a medida *betweenness* alcançou melhores resultados comparado às outras medidas de centralidade (grau, *closeness* e *clustering*) e seleção aleatória.

2) Observamos que se existir uma proporção equivalente de elementos rotulados de ambas as classes nas comunidades dos *datasets Digits1* e *USPS*, o desempenho do classificador pode ser maior. Porém, não conseguimos encontrar o mesmo padrão nos *datasets* g241c e g241d.

3) O pós-processamento desconectando vértices com classes diferentes não obteve resultados relevantes (comparando o experimento 1 e 2 a acurácia foi bastante próxima). O método também não mostrou nenhuma diferença significativa quando analisamos os resultados por proporção de elementos selecionados, com exceção da seleção com a métrica *betweenness* no *dataset USPS*.

4) Quando consideramos a complexidade computacional de cada medida discutido na Seção 3.6, observamos que apesar de ter os melhores resultados, a métrica *betweenness* também é a mais complexa se desconsiderarmos o pior caso do *clustering*. Além do *betweenness*, a outra métrica que conseguiu resultados positivos em alguns casos foi o *clustering*. Porém dependendo do grau dos nós do vértices, o *clustering* pode ser a medida mais complexa dentre as escolhidas no projeto ao considerarmos o seu pior caso.

7.2 Limitações do trabalho

Nesse trabalho foram utilizados apenas *datasets* binários, a fim de facilitar a identificação de padrões relacionados com a distribuição de elementos rotulados em comunidades.

Foram empregadas apenas quatro medidas clássicas de centralidade *degree*, *closeness*, *clustering* e *betweenness*.

Foram empregados apenas dois algoritmos de classificação semissupervisionados, contudo, em geral, não observamos resultados discrepantes entre eles.

Consideramos apenas uma técnica clássica de construção de grafos, o método kNN.

7.3 Trabalhos futuros

Podem ser explorados *datasets* multi-classes. Outras medidas de centralidades podem ser empregadas, bem como outras técnicas de construção de grafos e classificadores. Também podem ser feitos outros experimentos para tentar entender melhor o comportamento descrito no item 2 da seção 7.1.

O código está disponível no github do autor o qual poderá ser empregado na extensão do trabalho.

Referências

ARAÚJO, B.; ZHAO, L. Detecting and labeling representative nodes for network-based semi-supervised learning. In: IEEE. *The 2013 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2013. p. 1–8. Citado 3 vezes nas páginas 39, 40 e 41.

ARAÚJO, B.; ZHAO, L. Selecting nodes with inhomogeneous profile for labeling for network-based semi-supervised learning. In: IEEE. *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*. [S.l.], 2013. p. 428–434. Citado 2 vezes nas páginas 40 e 41.

ARAUJO, B.; ZHAO, L. Data heterogeneity consideration in semi-supervised learning. *Expert Systems with Applications*, Elsevier, v. 45, p. 234–247, 2016. Citado 3 vezes nas páginas 39, 40 e 41.

AYNAUD, T. *python-louvain x.y: Louvain algorithm for community detection*. 2020. https://github.com/taynaud/python-louvain. Citado na página 45.

BERTON, L. *Construção de redes baseadas em vizinhança para o aprendizado semissupervisionado*. Tese (Doutorado) — Universidade de São Paulo, 2016. Citado na página 43.

BLONDEL, V. D. et al. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, IOP Publishing, v. 2008, n. 10, p. P10008, 2008. Citado 2 vezes nas páginas 34 e 35.

CHAPELLE, O.; SCHÖLKOPF, B.; ZIEN, A. (Ed.). *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006. Disponível em: <http://www.kyb.tuebingen.mpg.de/ssl-book>. Citado 2 vezes nas páginas 44 e 45.

CHAPELLE, O.; SCHOLKOPF, B.; ZIEN, A. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, IEEE, v. 20, n. 3, p. 542–542, 2009. Citado 6 vezes nas páginas 19, 24, 27, 28, 29 e 43.

CHEN, J. et al. Big data challenge: a data management perspective. *Frontiers of Computer Science*, Springer, v. 7, n. 2, p. 157–164, 2013. Citado na página 19.

COHN, D.; ATLAS, L.; LADNER, R. Improving generalization with active learning. *Machine learning*, Springer, v. 15, n. 2, p. 201–221, 1994. Citado na página 38.

CORMEN, T. H. et al. *Introduction to algorithms*. [S.l.]: MIT press, 2009. Citado na página 32.

DASZYKOWSKI, M.; WALCZAK, B.; MASSART, D. L. Looking for natural patterns in analytical data. 2. tracing local density with optics. *Journal of chemical information and computer sciences*, ACS Publications, v. 42, n. 3, p. 500–507, 2002. Citado na página 37.

ELHAMIFAR, E.; SAPIRO, G.; VIDAL, R. See all by looking at a few: Sparse modeling for finding representative objects. In: IEEE. *2012 IEEE conference on computer vision and pattern recognition*. [S.l.], 2012. p. 1600–1607. Citado 3 vezes nas páginas 37, 38 e 41.

FACELI, K. et al. *Inteligência artificial: uma abordagem de aprendizado de máquina*. [S.l.]: LTC, 2011. Citado 8 vezes nas páginas 19, 23, 24, 25, 26, 27, 28 e 29.

FORTUNATO, S. Community detection in graphs. *Physics reports*, Elsevier, v. 486, n. 3-5, p. 75–174, 2010. Citado na página 34.

FRANCESCHET, M. *Local Clustering*. 2020. https://www.sci.unich.it/ francesc/teaching/network/clustering.html. Citado na página 34.

GRANDO, F.; NOBLE, D.; LAMB, L. C. An analysis of centrality measures for complex and social networks. In: IEEE. *2016 IEEE Global Communications Conference (GLOBECOM)*. [S.l.], 2016. p. 1–6. Citado na página 33.

HAGBERG, A. A.; SCHULT, D. A.; SWART, P. J. Exploring network structure, dynamics, and function using networkx. In: VAROQUAUX, G.; VAUGHT, T.; MILLMAN, J. (Ed.). *Proceedings of the 7th Python in Science Conference*. Pasadena, CA USA: [s.n.], 2008. p. 11 – 15. Citado na página 44.

HAN, J.; PEI, J.; KAMBER, M. *Data mining: concepts and techniques*. [S.l.]: Elsevier, 2011. Citado 2 vezes nas páginas 26 e 27.

HOLST, A. *Volume of data/information created worldwide from 2010 to 2024*. 2020. https://www.statista.com/statistics/871513/worldwide-data-created/. Citado na página 19.

HUBERT, L.; ARABIE, P. Comparing partitions. *Journal of classification*, Springer, v. 2, n. 1, p. 193–218, 1985. Citado na página 39.

HULL, J. J. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 16, n. 5, p. 550–554, 1994. Citado na página 38.

HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, IEEE Computer Society, v. 9, n. 3, p. 90, 2007. Citado na página 44.

KEMPER, A. *Valuation of network effects in software markets: A complex networks approach*. [S.l.]: Springer Science & Business Media, 2009. Citado na página 34.

KENNARD, R. W.; STONE, L. A. Computer aided design of experiments. *Technometrics*, Taylor & Francis, v. 11, n. 1, p. 137–148, 1969. Citado na página 37.

LEE, K.-C.; HO, J.; KRIEGMAN, D. J. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 27, n. 5, p. 684–698, 2005. Citado na página 38.

L'HEUREUX, A. et al. Machine learning with big data: Challenges and approaches. *IEEE Access*, IEEE, v. 5, p. 7776–7797, 2017. Citado na página 19.

MCKINNEY, W. et al. Data structures for statistical computing in python. In: AUSTIN, TX. *Proceedings of the 9th Python in Science Conference*. [S.l.], 2010. v. 445, p. 51–56. Citado na página 44.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072. Citado 2 vezes nas páginas 23 e 24.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: *Sistemas Inteligentes Fundamentos e Aplicações*. 1. ed. Barueri-SP: Manole Ltda, 2003. p. 89–114. ISBN 85-204-168. Citado 2 vezes nas páginas 23 e 24.

NENE, S. A.; NAYAR, S. K.; MURASE, H. object image library (coil-100. Citeseer, 1996. Citado na página 45.

NEWMAN, M. *Networks: An Introduction*. OUP Oxford, 2010. ISBN 9780199206650. Disponível em: <https://books.google.com.br/books?id=q7HVtpYVfC0C>. Citado 2 vezes nas páginas 32 e 33.

PARSAZAD, S.; SABOORI, E.; ALLAHYAR, A. Data selection for semi-supervised learning. *arXiv preprint arXiv:1208.1315*, 2012. Citado 3 vezes nas páginas 37, 38 e 41.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, v. 12, n. Oct, p. 2825–2830, 2011. Citado na página 44.

PEIKARI, M. et al. A cluster-then-label semi-supervised learning approach for pathology image classification. *Scientific reports*, Nature Publishing Group, v. 8, n. 1, p. 1–13, 2018. Citado 2 vezes nas páginas 38 e 41.

PROTOPAPADAKIS, E.; VOULODIMOS, A.; DOULAMIS, A. On the impact of labeled sample selection in semisupervised learning for complex visual recognition tasks. *Complexity*, Hindawi, v. 2018, 2018. Citado 2 vezes nas páginas 37 e 41.

SETTLES, B. *Active learning literature survey*. [S.l.], 2009. Citado na página 19.

TIPPING, M. E.; BISHOP, C. M. Mixtures of probabilistic principal component analyzers. *Neural computation*, MIT Press, v. 11, n. 2, p. 443–482, 1999. Citado na página 44.

TUR, G.; HAKKANI-TÜR, D.; SCHAPIRE, R. E. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, Elsevier, v. 45, n. 2, p. 171–186, 2005. Citado 2 vezes nas páginas 38 e 41.

WALT, S. V. D.; COLBERT, S. C.; VAROQUAUX, G. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, IEEE Computer Society, v. 13, n. 2, p. 22, 2011. Citado na página 44.

WASKOM, M. et al. *seaborn: v0.5.0 (November 2014)*. 2014. Disponível em: <https://doi.org/10.5281/zenodo.12710>. Citado na página 44.

YAMAGUCHI, Y. *Implementations of label propagation like algorithms*. 2020. `https://github.com/yamaguchiyuto/label_propagation`. Citado na página 44.

YOUNSI, R.; WANG, W. A new artificial immune system algorithm for clustering. In: SPRINGER. *International Conference on Intelligent Data Engineering and Automated Learning*. [S.l.], 2004. p. 58–64. Citado na página 37.

ZHANG, R. et al. Investigations on ensemble based semi-supervised acoustic model training. In: *Ninth European Conference on Speech Communication and Technology*. [S.l.: s.n.], 2005. Citado na página 38.

ZHANG, R.; RUDNICKY, A. I. A new data selection principle for semi-supervised incremental learning. In: IEEE. *18th International Conference on Pattern Recognition (ICPR'06)*. [S.l.], 2006. v. 2, p. 780–783. Citado 2 vezes nas páginas 39 e 41.

ZHOU, D. et al. Learning with local and global consistency. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2004. p. 321–328. Citado 2 vezes nas páginas 30 e 31.

ZHU, X.; GHAHRAMANI, Z. Learning from labeled and unlabeled data with label propagation. Citeseer, 2002. Citado 2 vezes nas páginas 29 e 30.

ZHU, X.; GHAHRAMANI, Z.; LAFFERTY, J. D. Semi-supervised learning using gaussian fields and harmonic functions. In: *Proceedings of the 20th International conference on Machine learning (ICML-03)*. [S.l.: s.n.], 2003. p. 912–919. Citado 2 vezes nas páginas 31 e 32.

ZHU, X.; GOLDBERG, A. B. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, Morgan & Claypool Publishers, v. 3, n. 1, p. 1–130, 2009. Citado 2 vezes nas páginas 19 e 27.